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

[1]North Carolina Agricultural and Technical State University

October 08, 2024

Posted on 8 Oct 2024 — CC-BY 4.0 — https://doi.org/10.36227/techrxiv.172840773.30369998/v1 — e-Prints posted on TechRxiv are preliminary reports that are not peer reviewed. They should not b...

# A Survey on Self-Running Networks: Concepts, Components, Opportunities, and Challenges

Shaghayegh Shajarian [1], Sajad Khorsandroo [1], and Mahmoud Abdelsalam [1].
[1]North Carolina Agricultural and Technical State University, USA

*Abstract*—**Over the past decade, the integration of the Internet of Things (IoT), 5th Generation Mobile Networks (5G), and Cloud 3.0 technologies has significantly transformed the internet landscape. These advancements have led to a massive increase in connected devices and data generation. Traditional network management, reliant on manual processes and simple mathematical models, has become insufficient due to the increased complexity and dynamic nature of modern networks. To address these new challenges, self-running networks as an enhanced network management approach can be utilized. These networks are characterized by their autonomous and intelligent capabilities that remove the need for human intervention. This paper explores the core components and functionalities of self-running networks and proposes their architecture. Additionally, the paper highlights ongoing research toward self-running networks. Current challenges in implementing self-running networks, such as security, interoperability, orchestration, and complexity, are also discussed. We complete this paper by examining use cases for two types of networks—Data Center Networks (DCN) and Wireless Sensor Networks (WSN)—and illustrate how self-running networks can revolutionize their functionality and efficiency.**

*Index Terms*—**Self-Running Networks, Network Management, Data Center Networks (DCN), Wireless Sensor Networks (WSN).**

## I. Introduction

The Internet has undergone a remarkable evolution over the past decade with the emergence of new technologies such as the Internet of Things (IoT), 5th Generation Mobile Network (5G), and cloud 3.0. The integration of the IoT has significantly expanded the number of connected devices, generating vast amounts of data from various sources. Simultaneously, 5G technology has revolutionized connectivity, offering faster and more reliable data transmission, thereby enabling real-time applications and services. Cloud 3.0 has further enhanced data storage and processing capabilities, providing scalable and cost-effective solutions to manage ever-increasing data volumes. Consequently, the sheer volume of data flowing through networks has skyrocketed, leading to unprecedented demands for network performance, reliability, manageability, and security.

In the early stages of computer networks, network management primarily relied on manual processes. This was because networks were relatively static, and they needed human intervention for tasks such as configuration, monitoring, and troubleshooting. However, as network complexity expanded in the late 20th century and the demand for optimal performance, security, and reliability increased, the traditional manual approach to network management faced significant challenges. These challenges stemmed from the heavy reliance on mathematical network models that considered the network status, actions, and performance feedback at the level of individual devices and protocols as their input [108].

In 1989, Artificial Intelligence (AI) technology was utilized in computer networks to improve the end-to-end Quality of Service (QoS) and Quality of Experience (QoE) [111], [6]. However, due to the need for powerful computational resources and the complexity and dynamicity of networks, using early AI-based algorithms did not make an improvement in QoE and QoS.

In 2001, IBM introduced the autonomic computing project known as eLiza, inspired by the autonomic nervous system. The primary objective of this project was to develop self-managing systems for computer networks, freeing administrators from mundane low-level management and operational tasks [39]. The concept of autonomic networks aimed to create self-aware, self-configuring, self-healing, and self-optimizing networks, reducing the need for constant human intervention in network management [59]. However, this promising management paradigm faced challenges that hindered its widespread adoption. One of the main obstacles was the complex technical implementation process, which made it difficult to realize the full potential of autonomic networks. Additionally, transitioning existing systems to this new paradigm posed significant challenges [32]. Despite these hurdles, the autonomic computing project laid the foundation for further research and development in the area of network automation and self-management.

In the late 2000s, the concept of programmable networks led to the emergence of Software-Defined Networking (SDN), which sought to simplify network management by separating the control and data planes.

The history of programmable networks can be divided into three parts: The **first** stage was active networking, which introduced programmable functions, the separation of the data plane and control plane, and OpenFlow Application Programming Interface (API). In the active network, data-plane programmability was more focused, but it did not offer practical performance and security. Also, after increasing traffic volumes, conventional routers and switches encounter different problems, including debugging configuration problems or controlling routing behavior due to the integration between the control plane and the data plane. Hence, in the **second** stage, which is the stage of control-data plane separation, the focus was on control plane programmability and innovation for network administrators. In the **third** stage, the OpenFlow switch that has a table of packet-handling rules identifies the matching rule [35].

Within this software-driven architecture, the network intelligence is centralized in the control plane. Consequently, network devices have undergone a shift, functioning as the data plane. These devices are programmable and can be easily configured and controlled through accessible interfaces such as ForCES, OpenFlow, and other similar protocols [76], [100]. However, due to the adoption of programmable network technologies, the network becomes increasingly dynamic, which needs monitoring of the network after-policy deployment to prevent misconfiguration [52].

Hence, they still require an occasional operator or external system intervention. These interventions are also necessary to define operational guidelines and provide information about the network's purpose and service instances. During the mid-2010s, it became evident that configuring networks solely on a device-to-device basis and at a low level was insufficient. Networks need to be designed to meet specific service requirements and support their unique features. This led to the emergence of Intent-Based Networking (IBN), which builds upon the foundation of SDN and introduces a higher level of intelligence [61]. IBN revolves around the concept of ”intent,” which is a set of operational guidelines and goals. An IBN is a network that autonomously operates and manages itself based on predefined intents. This approach empowers network operators to concentrate on their desired outcomes without becoming entangled in the complexity of low-level device configurations necessary to achieve those goals [81]. The capabilities of AI and Machine Learning (ML) in an IBN environment allow for the translation of intent into concrete actions that effectively lead to the desired outcomes. As a result, manual processes are significantly reduced, improving overall network efficiency [81]. Nevertheless, IBN did not adopt a fully decentralized model. Certain functions still require centralization [21].

Therefore, a novel approach to network management has emerged, opting for data-driven, ML-based models over traditional methods. The ultimate goal of this approach is to eliminate the need for human intervention in the management control loop [34]. Furthermore, the rapid progress in computing technologies, especially the development of Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs), has opened up exciting possibilities for applying ML techniques, such as deep neural networks, within the networking domain [97], [94]. This trend signals the inception of networks that function autonomously, continuously measuring, analyzing, and controlling themselves, giving rise to self-driving networks (aka self-running networks) [58]. Similar to self-driving cars, by removing the reliance on human operators, networks gain the ability to configure, correct, and explore solutions autonomously [44]. A good case in point is what Huawei introduced at Mobile World Congress 2018. By creating a digital replica between physical networks and business goals, they facilitate the transition of networks from SDN to autonomous driving networks [45]. Furthermore, Juniper Networks has introduced Mist AI [55], which leverages a combination of AI, ML, and data science techniques to optimize user experiences and simplify operations across wireless access, wired access, and SD-WAN domains. Their system gathers data from various sources, including Juniper Mist access Points, switches, smart routers, and firewalls, to gain end-to-end insights into user experiences. This includes automated event correlation, identification of root causes, self-driving network operations, network assurance, proactive anomaly detection, and other autonomous capabilities. Fig 1 presents a chronological representation of the developments in the way of the self-running networks.

The remainder of the paper is organized as follows: Section II defines self-running networks and outlines their key components. In Section III, we discuss the four main functionalities of self-running networks, propose an architecture, and examine each layer within it. Section IV reviews the current research and developments in the field. In Section V, we address the challenges and considerations involved with self-running networks. Section VI presents two case studies to demonstrate the application of self-running networks. Finally, we conclude the paper with a brief summary of our work.

## II. Components of Self-Running Networks

With networks evolving to be more heterogeneous, complex, and dynamic, network management activities—such as performance, security, configuration, and fault management—pose more significant challenges. For instance, fault management involves various procedures to identify, isolate, and resolve abnormal network conditions, starting with data labeling [88]. However, labeling data manually and having human-made decisions incur high costs due to the substantial amount of human effort required.[99].

Self-running networks, also known as self-driving or autonomous networks, are advanced and autonomous network infrastructures that integrate telemetry, workflow automation, and Al/ML to enable proactive and adaptive network management [52] [55]. It eliminates the need for constant human intervention by accurately predicting changes, adapting to user behaviors, and optimizing network operations.

In this network, all parts and nodes can be self-protected, self-configuration, self-healing, and self-optimized using collected data [108]. It can self-configure, monitor, manage, correct, defend, and analyze without any human involvement [23], [1]. By leveraging predictive and adaptive capabilities, the self-running networks optimize the end-user experience and align network operations with the business goals. This paradigm is crucial in transitioning from 4G to 5G and beyond, as it facilitates intelligent functionality across all devices and applications [85], [108].

As shown in figure 2, the self-running networks must contain the following steps. The first step in creating an autonomous network system involves ”Interpreting High-Level Intents to Determine Demand.” This means that the system must be able to understand complex requirements and objectives set by human operators or higher-level systems. By doing that, the network can understand what is needed from it regarding performance, resources, and outcomes. Once

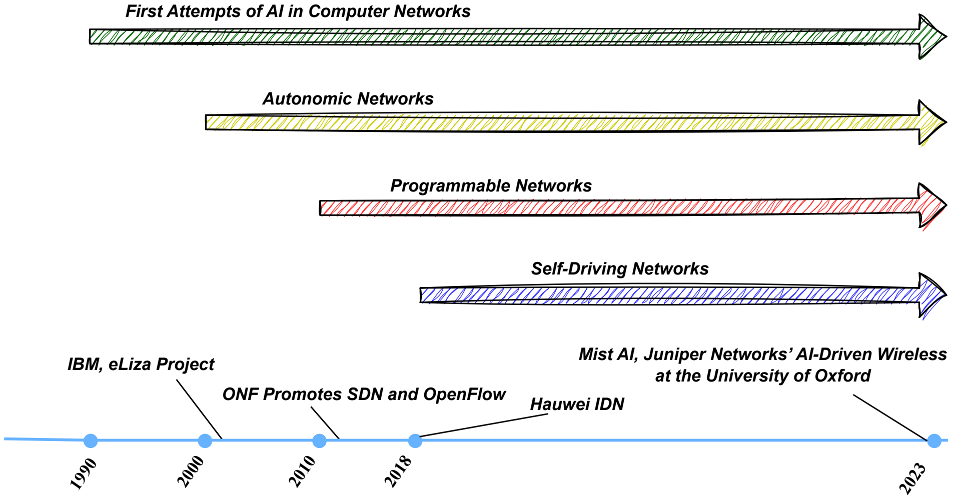
First Attempts of AI in Computer Networks
Autonomic Networks
Programmable Networks
Self-Driving Networks
IBM, eLiza Project
ONF Promotes SDN and OpenFlow
Hauwei IDN
Mist AI, Juniper Networks' AI-Driven Wireless at the University of Oxford
1990
2000
2010
2018
2023

Fig. 1: The Timeline to the Self-Running Networks

the network grasps the high-level intents, the second step is ”Network Self-Monitoring Aligned with Intent.” The network is expected to continuously monitor its performance and the state of its environment to ensure it operates according to the defined intents and adjusts as needed without external prompts. The third step, ”Predictive Analysis of Dynamic Data Patterns,” refers to the network's ability to anticipate future states by analyzing the data it collects. This predictive capability allows the network to proactively adjust to upcoming demands or potential disruptions by recognizing trends and patterns that may not be immediately apparent. The final step, ”Autonomous Networks Adaptation to Changes without Human Intervention,” emphasizes the need for the networks to independently implement changes in response to the dynamic conditions it has predicted or detected. With this level of autonomy, the network remains responsive and efficient without the need for constant human oversight, which reduces the operators' workload and improves the network's overall reliability [51].

The taxonomy of a self-running network is shown in Figure 3.

The aggregation of some components builds the self-running network. We can describe them as follows:

1) **Autonomic network management:** which contains self-configuration, self-protection, self-healing, and self-optimization. The network acts independently, making decisions based on its analysis and understanding of the environment. This level of autonomy reduces dependence on external instructions and manipulations and empowers the network to optimize its operations in real-time [55] [1]. The prediction of network metrics and helping the network administrator to understand the network's future can be the autonomic improvements of the network. In this regard, autonomic control loops are vital to self-running networks. They refer to closed-loop systems that enable the network to autonomously monitor, analyze, and adjust its operations based on predefined policies and objectives. In autonomic control loops, the network continuously collects data through telemetry and monitoring mechanisms to assess its current state. This data includes information on network performance, resource utilization, traffic patterns, and security threats.
2) **Telemetry:** Using telemetry techniques, the initial information describing the network states and containing details like traffic flow, network topology, policy rules, Netflow data, processing time, and link conditions are gathered. Hence, telemetry plays a pivotal role in self-running networks by providing comprehensive data on device states, customer experiences, and packet information. Analyzing the behavior of the networks has become crucial in network management aspects, including security, network performance, and fault management. This data enables the network to gain profound insights, make informed decisions, and optimize its performance and resource allocation [55].
3) **Automation:** It extends to automatic service placement, service motions, upgrades based on configured services, and network responses driven by ML. Increased data input into training algorithms enhances the network's intelligence and efficiency in managing its own operations[75]. The network intelligently reasons about its current state, interprets the available information and provides recommendations for re-configurations. Network automation becomes possible using SDN and Network Functions Virtualization (NFV) [37].
4) **Decision-making:** In self-running networks evolve from static rule-based systems to dynamic algorithms that learn from data inputs, make predictions, and take appropriate actions. Machine learning algorithms enable the network to analyze vast amounts of data, leading to more accurate decision-making and optimization of network performance [55]. In fact, automatic decision-making in 5G architecture should integrate with big data platforms to manage logs, traces, and configuration data from the

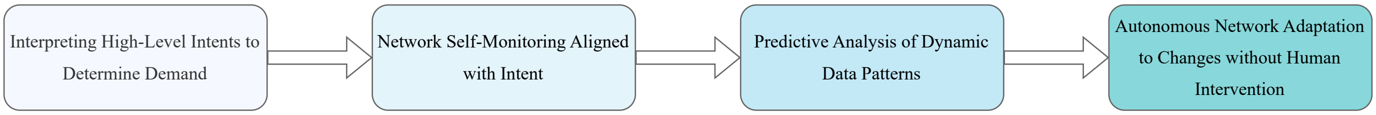
Interpreting High-Level Intents to Determine Demand
Network Self-Monitoring Aligned with Intent
Predictive Analysis of Dynamic Data Patterns
Autonomous Network Adaptation to Changes without Human Intervention

Fig. 2: Main Steps in Building Self-Running Networks

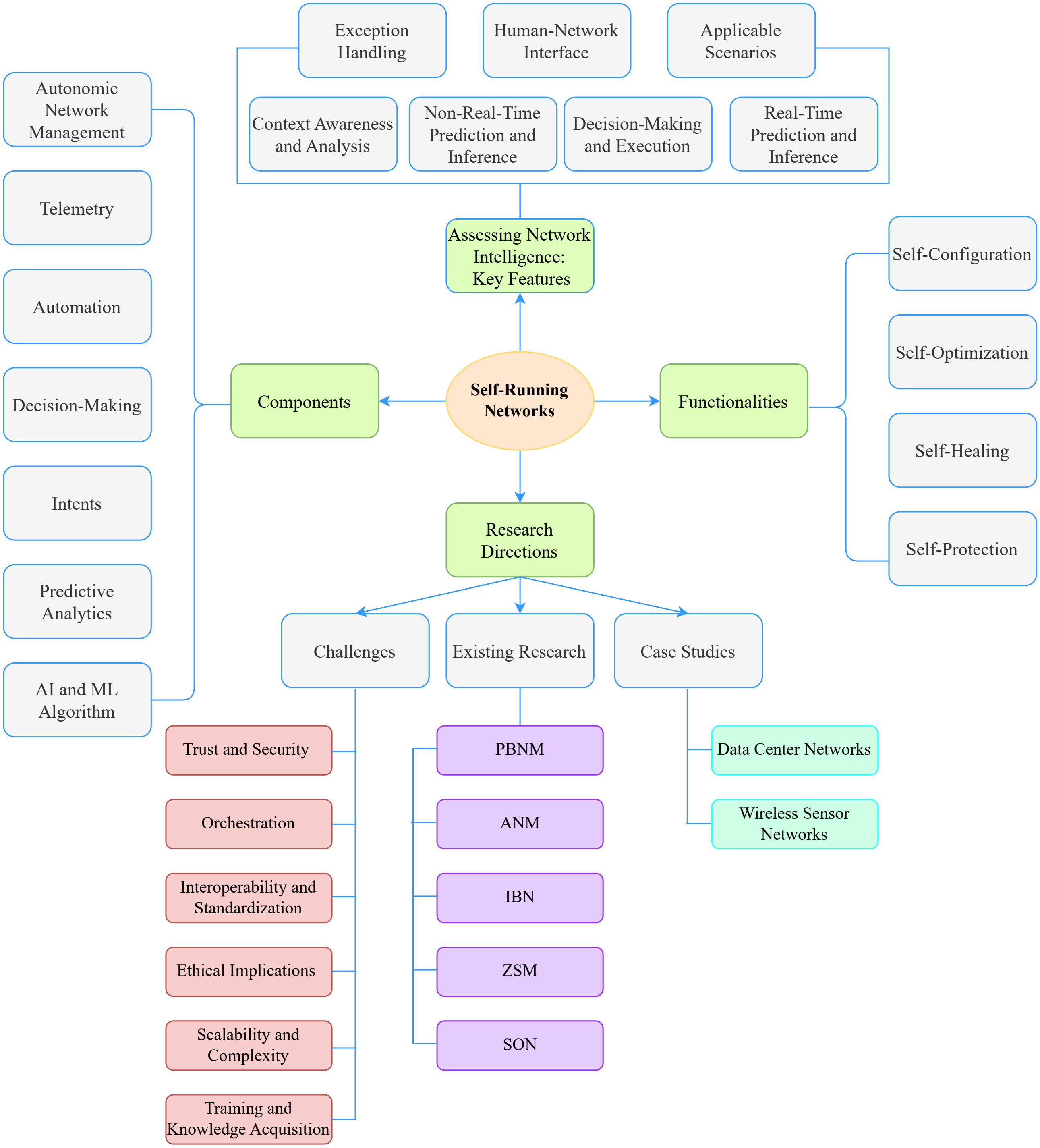

Fig. 3: Taxonomy of Self-Running Networks

network.

5) **Intents:** Intents refer to the high-level goals and objectives that network administrators or operators have for the network's behavior and performance, and IBN is a concept that aims to simplify network management and operation by allowing administrators to express their intentions in a more human-readable and business-oriented manner, rather than dealing with low-level technical configurations. IBN represents a significant milestone in the development of self-running networks. It allows operators to specify high-level policies and goals for the network without delving into low-level implementation details. IBN abstracts the complexity of network management, enabling operators to focus on desired outcomes while the network autonomously implements and adapts to achieve those goals [21] [52].

   High-level intents related to QoS, performance, and security should be considered without considering low-level details for implementing these intents. In this case, by using Natural Language Processing (NLP), the operators can use natural language to specify intents. After deploying intents and collecting data, the network can leverage ML-based techniques to make autonomous decisions[51].
6) **AI and ML Algorithms:** Creating cognitive and intelligent networks using ML techniques. Since the black-box nature of ML-based algorithms, this can be a solution for most challenges regarding self-running networks. For instance, increasing unstructured unlabeled data makes labeling data manually costly since it requires much human effort. By implementing an ML-based Network Management System (NMS) to predict anomalies, faults, and bottlenecks based on the environment and its requirements, the network can learn from data and eventually optimize, control, and manage itself and adapt to sudden changes[97].
7) **Predictive Analytics:** AI and ML models analyze historical data to forecast potential network problems like congestion, security risks, or performance decline. By identifying these issues before they escalate, self-running networks can proactively implement measures to address and prevent them.

## III. Functionalities of Self-Running Networks

The functionalities of self-running networks are defined as self-* that contain self-configuration, self-optimization, self-protection, and self-healing.

### *A. Self-Configuration*

Self-configuration represents a fundamental shift in deploying and managing network elements such as routers and switches. In traditional networks, configuring devices and deploying new services require repetitive but careful manipulation that relies on manual input from operators.

Despite careful handling, this process can inadvertently result in unforeseen consequences, leading to errors, inconsistencies, and time-consuming procedures [41]. Nevertheless, self-configuring networks can configure themselves using automation and intelligent algorithms based on high-level specifications. More specifically, when a new network element is introduced, it communicates its capabilities and prerequisites. The network's autonomous systems analyze this information and determine the most fitting configuration. This process includes assigning IP addresses, defining network topologies, establishing communication protocols, and ensuring interoperability. By having self-configured networks, these networks significantly reduce human intervention, minimize configuration errors, and accelerate the deployment of new devices and services in the networks. Although self-configuration addresses traditional networks' challenges, it can encounter new challenges. For example, the previous AI-based algorithms would be ineffective when new patterns appear in the network. Hence, an ML algorithm should be updated based on new patterns. [79].

### *B. Self-Optimization*

Self-optimization empowers networks to adapt and fine-tune their performance based on changing conditions and user demands. In conventional networks, administrators manually adjust settings like bandwidth allocation, QoS parameters, and routing paths to optimize performance. By implementing self-optimization, sophisticated algorithms continuously analyze network data, including real-time traffic patterns, resource allocation, and application requirements.

By leveraging AI-based methods and predictive analytics, these networks make decisions on adjusting parameters to optimize performance automatically [16]. For instance, based on the network situation, the networks may involve reallocating bandwidth to critical applications during peak usage, optimizing routing paths to minimize latency, or adjusting power levels to enhance energy efficiency. The result is a network that can anticipate and respond to fluctuations in demand, ensuring consistently high performance and efficiency while minimizing human supervision.

### *C. Self-Healing*

Networks with self-healing functionality are often called ”ad hoc” networks. These networks can heal disruptions arising from various factors, such as the ongoing movement of nodes, shifts in radio frequency propagation, physical node degradation, and other pertinent variables [29]. More specifically, a network's self-healing capability addresses the issues posed by failures, attacks, and disruptions in the networks. In conventional networks, human intervention is required to identify the problem, diagnose the cause, and implement a solution when a component fails. Self-healing functionality, on the other hand, proactively monitors networks. When internal and external faults and attacks arise, such as a hardware malfunction or a sudden surge in traffic causing congestion, the network with self-healing functionality can detect the anomaly, isolate the affected area, and reroute traffic to ensure continuity of service [107]. In this way, the network can repair or replace faulty components without manual intervention. This level of intelligence leads to improved reliability and reduced time and effort, which are needed in this process.

### D. Self-Protection

Self-protection is an essential aspect of network security that keeps networks against various cyber threats and attacks. Traditional security measures often involve static rules and manual responses to threats. By utilizing self-protection functionality, the networks proactively monitor traffic for revealing security risks, e.g., abnormal behavior, unauthorized access attempts, and potential vulnerabilities. When suspicious activity is detected, the network's self-protection takes immediate action to prevent failures, such as isolating compromised devices, blocking malicious traffic, and applying security patches. Additionally, because of the utilization of Ml-based algorithms, the networks can adapt and evolve their defense mechanisms in response to emerging threats.

The four mentioned functionalities of self-running networks —self-configuration, self-healing, self-optimization, and self-protection are the keys to a new era of network management and operation. These capabilities utilize AI-based methods to create networks that are adaptive, reliable, efficient, and secure. By removing human intervention and leveraging real-time insights, self-running networks improve overall network performance and enhance user experiences.

### E. Architecture:

Toward the goal of having self-running networks, in this section, the unified definition of the architecture of these networks will be provided. This architecture should represent the intelligence of the whole network. In this regard, the need for metrics to evaluate the network intelligence is becoming important, which leads to having a clear insight into the network autonomy and intelligence level.

In 2014, the Society of Automotive Engineers (SAE) suggested the definition of automatic driving levels [86]. Borrowing this idea, in 2020, the International Telecommunication Union (ITU) proposed six levels of network intelligence from level 1, which is based on human roles, toward level 6, which is the interaction of all layers as a whole system. To understand the intelligence that the system needs at each level, seven key features can be considered [48].

**Seven Key Features to Assess Network Intelligence Levels:**

1. **Context Awareness and Analysis:** This feature uses a large dataset to be aware of events within the network and identifies their root causes. They are the first step toward network automation since they do not need real-time analysis [48] [47].
2. **Non-Real-Time Prediction and Inference:** After implementing the first feature using AI/ML, predicting potential events or changes is the next step toward an intelligent network, which can be achieved in a non-real-time scenario [48] [47].
3. **Decision-Making and Execution:** Based on predictions from the previous level, the network can autonomously establish optimization policies and execute based on them. Hence, until this level, all autonomy is based on non-real-time scenarios [48] [47].
4. **Real-Time Prediction and Inference:** This is focused on swift predictions, which need to have real-time analysis accompanied by immediate actions [48] [47].
5. **Exception Handling:** This deals with the management and recovery of sudden events. For exception handling, the network does not care about the current rule and decides based on the system's needs and changes things [48] [47].
6. **Human-Network Interface:** This feature considers how operators interact with the network. The interaction can be through traditional signaling or command interactions or via an intent [48] [47].
7. **Applicable Scenarios:** This evaluates the scope of intelligence of the network, whether they address specific services or scopes or encompass scenarios throughout the entire network [48] [47].

Once a comprehensive understanding of the intelligent network is attained, it becomes possible to establish a well-defined architectural framework. Hence, each AI subsystem can be deployed with the goal of attaining the highest level of network intelligence. It is worth noting that implementing intelligence at each level can affect different parts of the architecture.

The high-level view of the proposed architecture, as depicted in 4, has four layers:

The **Sensing Layer** contains all sensors and smart devices and has the ability to collect data and monitor the environment. The network states, such as traffic data, network topology, link status, and resource status, are collected with telemetry techniques from the first layer. Also, this layer has Edge devices and edge servers, which are useful for local processing. By performing computation on data collected from end devices and network elements, latency and bandwidth usage on the core network will be reduced. There are three main connections in this layer:

- *Network Element to End Device:* Network elements facilitate the connection of end devices to the network by providing pathways for data to travel.
- *End Device to Edge Server:* End devices send data to edge servers for processing. Edge servers can also send data and control signals back to end devices to provide services or perform actions with low latency.
- *Edge Server to Network Element:* Edge servers interact with network elements to send processed data to the core network or to receive data from it.

Then, utilizing network elements, e.g., base stations, routers, and switches, the data is sent to the **Data Processing Layer**, which has data centers and computational servers. In this layer, the monitoring component receives data; then, the collected data is sent to the analysis component, which can be useful in identifying the changes in the network and predicting future changes. After that, based on the result of the analysis, the corresponding intent will be selected, and at the end, the network will change based on the situation, which can be a reconfiguration of the network (self-configuration), preventing failures (self-protection), or fixing faults (self-healing). The intents and the result of this process will be saved as knowl-

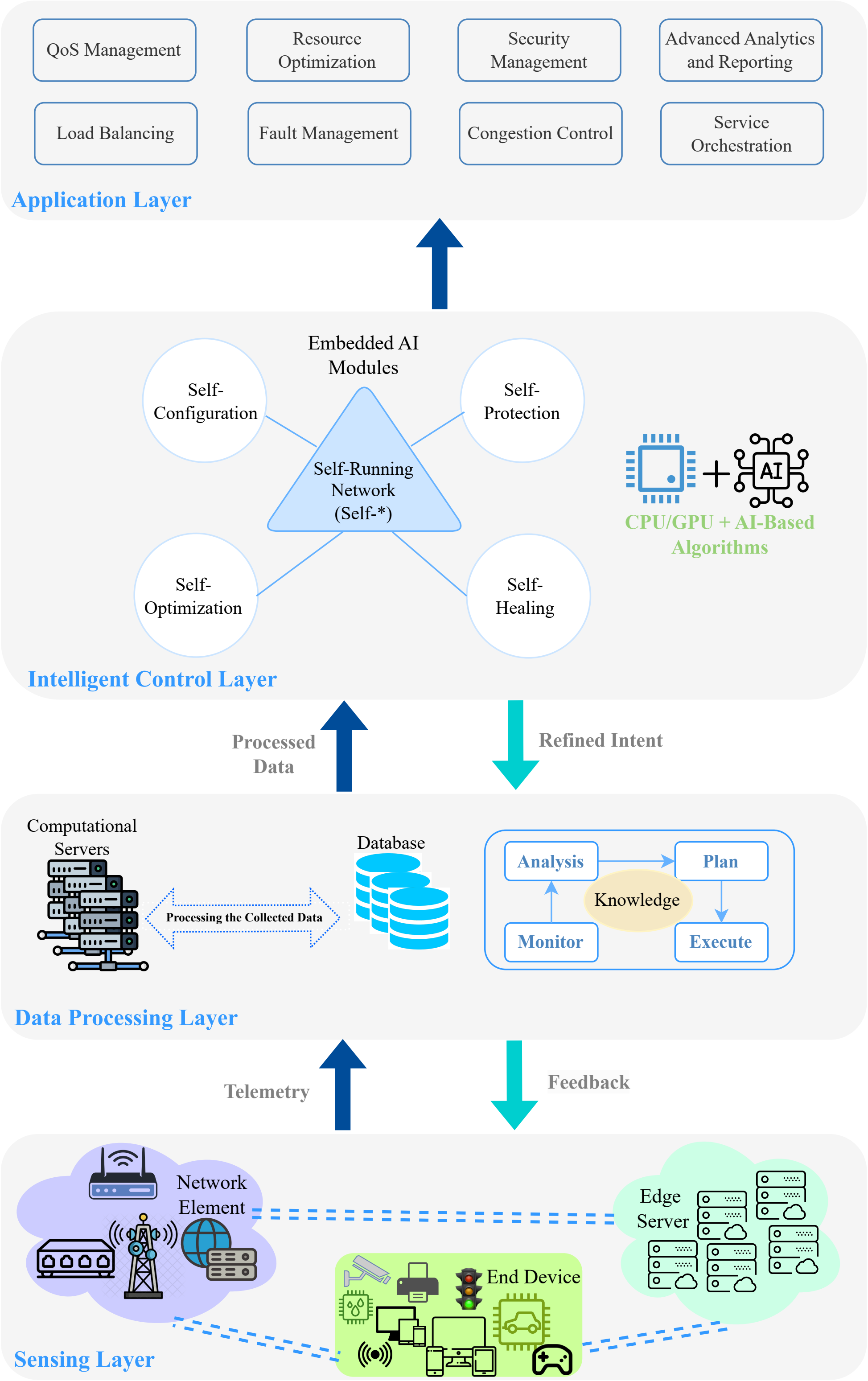

Fig. 4: The Architecture for Self-Running Networks

edge based on MAPE-K reference model [14]. Moreover, by utilizing a database, this layer maintains the unprocessed data collected from networks and instances for the purpose of training, so it does pre-processing to feed AI modules.

The next layer, which is the core of the intelligence and autonomy of self-running networks, is the **Intelligent Control Layer**. This layer consists of self-* functionalities, which are self-configuration, self-optimization, self-healing, and self-protection. Since this layer has the most computational and memory resources, it has the ability to do resource-intensive tasks such as resource management, intent interpretation, QoS management, and failure management. For instance, in self-configuration, AI-based modules are deployed to configure network parameters based on changes and modifications. In addition, telemetry can help to have real-time packets and network state data, which can be useful for proactive prediction. So, the intelligent control layer is responsible for learning, predicting, and regenerating the intents based on the result of AI-based modules. The output of AI-based algorithms will be fed into applications or use cases such as fairness, congestion control, and load balancing. Based on the output, the intent becomes revised and fed back to the data processing layer[69].

Finally, the **Application Layer** contains applications that would gain benefits in self-running networks, such as load balancing, network congestion, and security management [102].

It is worth mentioning that in a self-running network, the quality of data is very important, so we should design a network that can improve the quality of data that is used for training AI-based algorithms. Hence, providing appropriate intent that contains users' expectations, network goals, application functions, and services would be considered.[34].

## IV. Research and Development Towards Self-Running Networks

This section will explore the progression toward self-running networks, which considers two subcategories, paradigms and applications, from initial attempts to contemporary studies. Table I.

### A. Paradigms

**Policy-Based Network Management (PBNM)** The framework and architectural components of PBNM were defined by the Internet Engineering Task Force (IETF) in 2000 [24][40].

PBNM is an approach for network management 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 have the ability to develop and implement these policies, and they are responsible for identifying and setting up appropriate policy configurations [40]. In general, these policies can be categorized into two groups: authorization policies, which determine what activities a user has permission to engage within the system, and obligation policies, which define actions that the system must or must not take in response to specific events [90].

Verma [98] presents a framework for easing the complexities of managing modern IP networks through policy-based management. 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. By implementing centralization and business-level abstractions, the framework reduces the manual effort required for network management. 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. [82] utilize PBNM in their study of Home Area Network (HAN). HAN, often involves wired and wireless devices—including home appliances, gaming consoles, and cameras. Managing such networks can be challenging due to their complexity and the number of smart devices connected. 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.

In another study, Solomon et al. [91] propose a Policy Creation Model for managing network bandwidth at the Botswana International University of Science and Technology (BIUST). By employing a structured approach to policy creation and implementation, the research aims to optimize 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. It improves network efficiency and reduces non-critical bandwidth usage. 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. [9] 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 MITM attacks. However, while the system includes mechanisms for detecting malicious behavior, it does not discuss the potential privacy implications of monitoring and analyzing user activities. Hence, ensuring compliance with privacy regulations and protecting user data while maintaining security is a significant challenge that is not addressed.

**Autonomic Networks Management:** In a heterogeneous and complex networking ecosystem, users should be able to have contextualized, proactive, and personalized access to services everywhere. To achieve this target, the management of networks cannot be done on a device-to-device basis and with low-level configuration since the networks should be aligned with the specific requirements of the services and the service features they are intended to support. Autonomic Networks Management (ANM) addresses the ability of networks to be aware of themselves and their environment[59].

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 [13]. With respect to ANM, the ultimate aim is to create self-managed networks to overcome the rapidly growing complexity of networks. In 2001, IBM presented the autonomic computing framework, describing a system with 'self-x' properties, such as self-healing, self-configuration, self-optimization, and self-protection [39]. 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 [92].

In this scenario, the autonomic network comprises a multitude of 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 [59]. The vision towards autonomic manager includes the following closed-control loops: sensing (or monitoring) changes in the network and its environment; analyzing changes to achieve the goals; planning reconfiguration if goals cannot be achieved; executing those changes; and observing the results. The operation of the control loops is enhanced by adding learning and reasoning processes, as well as by employing a well-structured knowledge base [32] [46].

In this topic, several papers focus on the architecture of autonomic networks [20], [53], [27], [33],[18],[67],[13]. For example, in the study by Arzo et al. [13] introduces an architecture designed to autonomously manage complex networks by using multiple interacting agents. Each agent performs specific network functions autonomously, such as network slicing, path computation, and QoS monitoring. The architecture aims to replace traditional monolithic network management systems with a more flexible and scalable approach.

Multiple papers focus on integrating autonomic network management with emerging network technologies to address the increasing complexity of network systems (including [68], [95], and [54], among others). For instance, Tsagkaris et al. [95] aim to enhance network management by integrating ANM and SDN to develop a customizable management framework. This approach addresses the increasing complexity of network systems by leveraging ANM's self-governing capabilities and SDN's programmable network control. 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. [54] focus on autonomic network management in 5G systems. Like the other papers, it leverages technologies such as SDN and NFV to propose a management framework. The ultimate goal of this study is to simplify network management, enhance efficiency, and reduce operational costs. The presented framework developed under the EU H2020 SELFNET project emphasizes self-healing, self-protection, and self-optimization capabilities 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. [92] 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 to ensure robust network operations.

Despite the aim of autonomic networks to achieve self-management, they are unable to eliminate the necessity for operator or external system intervention completely. This is because autonomic networks require an operator or outside system to define operational guidance and information regarding their purposes and service instances [21].

**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 the evolving of 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 that they can be defined across different technologies [109]. 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 [21].

In the industry part, in 2017, Cisco introduced the IBN system with three main components: translation, activation, and assurance. One year after that, Huawei introduced Intent-driven networks (IDN), which contains a Network cloud engine for having intelligent network control and management. Prominent industry leaders like Cisco, Huawei, and the IETF have spearheaded the development of IBN systems. Hence, one of the important processes in IBN is the translation of intent to configuration, which relies on ML or AI algorithms.

Multiple studies demonstrate how IBN principles are implemented in different contexts and for various purposes within 5G networks [2], [3] [22] [77] [10] [112] [73] [96] [70] [4] [89] [72] [50] among others.

For example, Abbas et al. [2] leverage an IBN-based approach for performing end-to-end network slicing (slicing the RAN and core network) 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 proactive, autonomous actions for both domains. Following this work, the authors focus on an IBN approach for managing the lifecycle of network slices in 5G networks. The proposed system automates the creation, configuration, and management of network slices using high-level intent expressions [3].

Collet et al. [22] introduce LossLeaP, a deep learning-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 [74] [77] [84]. For instance, in a recent study, Orlandi et al. [77] implement an architecture of IBN, particularly to simplify user interactions and service configurations through NLP and user-friendly interfaces.

Hence, IBN management is more about aligning network operations with business intents and continuously 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.

**Zero-Touch Network:** The European Telecommunications Standards Institute (ETSI) established Zero-touch network and Service Management (ZSM) in December 2017 with the primary goal of fully automating networks and moving away from inflexible management systems towards more adaptable services [38], [30]. 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 [66]. This architecture should be modular, extensible, scalable, and resilient to failure. This closed-loop management automation reduces the risk of human error and OPerating EXpenses (OPEX) and improves the flexibility and efficiency of services [17].

Multiple research papers were published regarding ZSM. Rezazadeh et al. [83] 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. [25] present a novel methodology for end-to-end service monitoring in zero-touch networks by introducing the Monitoring Model Generator (MMG) component. The MMG uses service deployment models and standard information models to create high-level monitoring templates based on an ontology-based schema. The proposed approach is validated through a proof of concept implementation.

A study by Angui et al. [12] 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 have secure data-sharing channels in ZSM, Kumar et al. [60] propose a new secure framework using blockchain and deep learning algorithms. To be more specific, they propose Intrusion Detection Systems (IDS), which are a combination of Variational AutoEncoder and Attention-based Gated Recurrent Units (AGRU). By using Variational AutoEncoder, they are free from having domain-specific knowledge for extracting features, and by implementing AGRU, they detect intrusion.

### *B. Applications*

In this section, we focus on state-of-the-art research on the application of self-running networks.

**Resource Management** As the traditional methods for resource management, such as using heuristics, do not work in today's complex networks, the researchers focus on how resource management can be autonomous.

Zhang et al. [110] propose an ML-based framework to enhance radio resource management in Non-Orthogonal Multiple Access (NOMA) millimeter-wave, 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 quality of service QoS, interference limitations, and power limitations. The framework involves solving the user association problem using the Lagrange dual decomposition method, while subchannel

TABLE I: Existing Research Toward Self-Running Networks Paradigms

| Category | Paper/Year | Objective | Contribution | Limitation |
|---|---|---|---|---|
| **Policy-Based Network Management** | Verma 2002 [98] | Simplifying the management of complex network infrastructures by using policy-based management frameworks | • Developed PBNM that simplifies network administration by centralizing configuration and using business-level abstractions. | • It does not address the potential latency and performance bottlenecks that may arise from centralizing policy management in dynamic and large-scale networks. |
| | Rana et al. 2009 [82] | Aims to use 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 in HAN<br>• Showed significant enhancements in VoIP quality and reduction in packet loss through policy enforcement | • Real-world networks can experience varying conditions and user behaviors, which might challenge the static policies used in the study. |
| | Solomon et al. 2017 [91] | Proposing the adoption of 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.<br>• Demonstrated improved bandwidth management and network efficiency post-policy implementation. | • The policy management system lacks dynamic reconfiguration capabilities, which can lead to suboptimal performance in environments with variable traffic patterns and usage needs. |
| | Alquhayz et al. 2019 [9] | Focuses on developing a policy-based approach to security management systems to prevent end-user devices from being used as attack tools | • Development of a policy-based security management system that integrates with the Y-Comm architecture to enhance security in 5G networks.<br>• Introduction of an intelligent agent mechanism to detect malicious behavior in end-user devices. | • While the system includes mechanisms for detecting malicious behavior, it does not discuss the potential privacy implications of monitoring and analyzing user activities |
| **Autonomic Networks Management** | Arzo et al. 2021 [13] | Propose a novel architecture for autonomic network management using a multi-agent system | • Proposing a multi-agent-based autonomic network management architecture and Conducting an evaluation of the system's performance in terms of functionality, reliability, latency, and resource consumption. | • There is no detail about practical integration challenges with existing systems. |
| | Tsagkaris et al. 2015 [95] | Aims to enhance network management by integrating ANM and SDN | • Developed a customizable ANM framework integrating SDN/OpenFlow capabilities. | • lacks extensive real-world validation, which limits the understanding of its scalability and performance in operational networks. |
| | Jiang et al. 2017 [54] | Design and implement an autonomic network management framework for 5G mobile networks | • Implementing framework under the EU H2020 SELFNET project, a novel autonomic management system for software-defined and virtualized 5G networks<br>• Designing network intelligence through ML techniques for self-healing, self-protection, and self-optimization. | • The current implementation demonstrates traffic congestion scenarios and the relevance of specific network metrics which identifies the need for more comprehensive testing. |
| | Stamou et al 2019 [92] | Aims to enhance device-to-device communication and resource management in cognitive radio networks | • Proposing a framework to integrate ANM with SDR,SDN, and NFV<br>• The framework is tested on real-world testbeds to demonstrate the feasibility and effectiveness of the proposed approach. | • The combination of SDR, SDN, and NFV introduces complex dependencies and potential attack vectors that can be exploited by malicious entities. |
| **Intent-Based Networks Management** | Abbas et al. 2020 [2] | Implementing an IBN slicing system for 5G networks that can slice and manage core network and RAN resources | • Implemented a GAN deep learning model for predictive resource management and enhancing slice resource assurance.<br>• Achieved automation of the lifecycle management of network services, significantly reducing manual effort. | • The reliance on open-source platforms and automated processes may expose the system to cyber-attacks, unauthorized access, and data breaches that pose significant risks to network integrity and user data protection. |
| | Abbas et al 2021 [3] | Developing an intent-based network slice lifecycle management system for 5G networks that can automate the creation, configuration, and management of network slices using high-level intent expressions. | • Proposing an IBN framework for managing the lifecycle of network slices in 5G networks.<br>• Implementing a prototype system and validating it with various performance tests to demonstrate its effectiveness and efficiency. | • The system has limitations in managing user mobility. |
| | Collet et al. 2022 [22] | Develop a forecasting model that autonomously learns the relationship between predictions and network management objectives to optimize complex, machine-translated goals in IBN | • Implementation of an architecture that allows the model to autonomously learn the most suitable loss function for any given objective, minimizing the need for manual intervention. | • The model includes a dual DNN architecture (predictor and loss-learning block) and employs techniques like co-training and cyclic learning rates, which introduce significant computational demands. |
| | Orlandi et al. 2024 [77] | Simplifying service ordering and configuration through user-friendly interfaces and NLP | • Framework for engaging non-expert users; NLP-enhanced chatbot for intent translation; vertical automation from service order to network deployment. | • The effectiveness of the system depends on the comprehensiveness and accuracy of the predefined product catalogs and metadata. |

TABLE I: Existing Research Toward Self-Running Networks Paradigms (cont.)

| Category | Paper/Year | Objective | Contribution | Limitation |
|---|---|---|---|---|
| **Zero-Touch Networks Management** | Rezazadeh et al. 2020 [83] | Developing an AI-driven, zero-touch network slicing solution using twin delayed deep deterministic policy gradient | • Adoption and fine-tuning of the TD3 method to enhance the convergence speed and stability of learning in continuous DRL tasks<br>• Development of a comprehensive 5G network slicing environment using the OpenAI Gym toolkit to enable standardized testing and comparison of different DRL algorithms. | • Implementation in real-world scenarios might encounter complexities not fully addressed in a simulated environment. |
| | Sousa et al. 2021 [25] | Enhancing end-to-end service monitoring | • Introduction of the MMG component that generates Service Monitoring Models (SMM) using service deployment models and standard information models. | • While validated in a proof of concept, the scalability and performance of the MMG in large-scale, real-world networks require further evaluation. |
| | Angui et al. 2022 [12] | Automating the deployment of Cloud-RAN in 6G networks, focusing on latency and resource management | • Introduction and validation of a ZTC model for Cloud-RAN that automates resource discovery, deployment, and configuration of network elements.<br>• Developed a protocol using Elliptic Curve Cryptography for session key establishment, and Proof-of-Authority for block verification | • While compliant with multiple deployment scenarios, the practical implementation might still face unforeseen challenges in different real-world settings. |
| | Kumar et al. 2022 [60] | Enhancing security in IoT-enabled Zero Touch Networks | • Created a novel intrusion detection system combining Variational AutoEncoder and Attention-based Gated Recurrent Units to automatically extract features and detect intrusions<br>• Developed a protocol using Elliptic Curve Cryptography for session key establishment, and Proof-of-Authority for block verification | • The current implementation is limited to testing with a maximum of 36 nodes and 304 transactions, indicating potential challenges in scaling up to larger, real-world applications. |

allocation and power control are addressed through semi-supervised learning and Deep Neural Networks (DNNs). By introducing an intelligent control center and utilizing a co-training semi-supervised learning algorithm for subchannel allocation and a DNNs for power allocation, they improve the approximation and generalization capabilities for subchannel allocation. The simulation results show that the proposed scheme enhances energy efficiency compared to traditional methods and has a lower computational complexity. The framework efficiently balances the load among base stations and optimizes resource allocation in real-time.

The paper by Yu et al. [106] proposes an intelligent ultradense edge computing (I-UDEC) framework to address the challenges of resource management in multiaccess edge computing within 5G ultradense networks. The framework combines blockchain and AI to optimize computation offloading, resource allocation, and service caching placement. To implement this framework, a two-timescale deep reinforcement learning (2Ts-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 2Ts-DRL 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 2Ts-DRL algorithm can reduce task execution time by up to 31.87% compared to other benchmark strategies. The framework's effectiveness is validated through simulations using the MATLAB RL toolbox. By leveraging FL, the framework also maintains user data privacy.

Mason et al. [71] work on a solution for dynamic resource allocation in Network Slicing scenarios using Deep Reinforcement Learning (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 (eMBB) and Ultra Reliable Low Latency Communication (URLLC). 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 significantly 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. [8] focus on resource allocation within the Random Access Network, utilizing deep multi-agent reinforcement learning. 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.

**Network Traffic Analysis and Prediction:** In the face of rapid traffic growth due to technological advancements, 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 dynamically adjust routes that enhance the user experience by reducing latency and packet 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 [31]. By fully harnessing the capabilities of self-running networks, it can detect subtle anomalies and variations in traffic that traditional methods might miss, thereby providing a more robust defense against cyber threats.

Xavier et al. [101] introduce a novel 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, Intrusion Detection Systems 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 with just a few packets, achieving up to 97% accuracy with only two packets in certain cases. Despite the limitations imposed by P4 and the hardware, MAP4 maintains acceptable performance, ensuring that the deployed models do not become network bottlenecks.

The technical study by Shahraki et al. [88] 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 Learning Active Learning, 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.

Hence, Active Learning can be integrated into self-running network management systems to enable automated classification and management of network traffic. This helps in real-time decision-making and reduces the need for constant human oversight. However, the computational overhead associated with Active Learning can be a significant limitation since these techniques require additional computations to identify and select the most informative samples for labeling, which can increase the processing time and resource usage.

Hardegen et al. [42] aim to 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 RNNs and GNNs. These alternative models might better capture sequential or structured data and offer improved performance and accuracy in predicting network flow characteristics.

**Routing:** Self-running networks, specifically in the context of routing, leverage advanced algorithms and machine learning to dynamically 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. This ensures that data flows through the most efficient paths, avoiding bottlenecks and minimizing delays. 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 capability is particularly 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. Moreover, self-running networks 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. [19] present RL-Routing, an innovative algorithm designed to optimize routing in SDN using deep reinforcement learning. 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. RL-Routing leverages a deep reinforcement learning approach that 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 Fat-tree, NSFNet, and ARPANet, demonstrate that RL-Routing outperforms traditional methods in terms of throughput and communication delay. The flexible reward function used in RL-Routing allows it to be efficient for optimizing either upward or downward network throughput and makes it highly adaptable to various network demands. Additionally, the algorithm addresses scalability concerns by requiring only one agent per switch, reducing the overhead and complexity typically associated with multi-agent systems.

## V. Challenges and Considerations of Self-Running Networks

### A. *Trust and Security Concerns*

Several studies show that ML-based models have the potential to be the targets of security and privacy attacks, such as poisoning training data or adversarial attacks that can damage networks [15]. For instance, when using deep learning-based

TABLE II: Existing Research Toward Self-Running Networks Applications

| Paper/Year | Objective | Contribution | Limitation | Methods |
|---|---|---|---|---|
| **Resource Management** | | | | |
| Zhang et al. 2020 [110] | Focuses on enhancing the performance and efficiency of resource management in NOMA networks | • Developed a novel DL-based framework for optimizing user association, subchannel allocation, and power allocation in NOMA mmWave heterogeneous networks<br>• The proposed method integrates semi-supervised learning to utilize both labeled and unlabeled data | • The effect of the DL algorithm is significantly dependent on the labeled samples, which are generated by iterative algorithms. | Lagrange Dual Decomposition, Semi-Supervised Learning and DNNs |
| Yu et al. 2020 [106] | Aims to achieve real-time and low overhead computation offloading decisions and resource allocation strategies | • Proposes a 2Ts-DRL approach for real-time computation offloading, resource allocation, and service caching placement in 5G ultra-dense networks.<br>• Leverages FL to train the 2Ts-DRL model to ensure data privacy for edge devices. | • The framework assumes homogeneous resource availability and does not fully address the complexity introduced by highly heterogeneous resources.<br>• It only assumes a static environment for the caching decisions. | Deep Reinforcement Learning, Federated Learning, Blockchain Integration |
| Mason et al. 2022 [71] | Aims to address the dynamic allocation of network resources in a Network Slicing scenario | • Developed a new resource allocation method using Deep Reinforcement Learning.<br>• Introduced a distributed architecture with multiple collaborating agents and demonstrated how transfer learning can enhance system performance. | • The system operates under partially observable conditions, where agents have limited views of the overall network status, which can affect decision accuracy.<br>• Comparisons are between the proposed method and static allocation and also empirical strategy. There is no comparison with other dynamic network resource allocation methods. | Deep Reinforcement Learning, Transfer Learning |
| Allahham et al. [8] | Focuses on malware detection, which locates code snippets, and on effectively explaining the decision of malware classifier | • Proposed effective deep learning approach for Linux malware detection<br>• Methodology detects malicious behavior in malware that uses inline assembly<br>• Explanation of malware classification results using LRP | • Lacks clear future technological guidance | Layer-wise relevance propagation (LRP) |
| **Traffic Analysis** | | | | |
| Xavier et al. 2022 [101] | Deploying ML models for traffic classification within programmable network devices | • Introduced the MAP4 framework to deploy ML-based traffic classification directly within network devices<br>• Validated MAP4 in real-world scenarios, including Intrusion Detection Systems and **IoT !** (**IoT !**)device classification, demonstrating high accuracy and minimal performance degradation, even at high transfer rates. | • The framework is tested on Netronome SmartNICs, which have constraints in terms of throughput, memory size, and processing power, limiting the applicability to edge rather than core network deployments. | Decision Trees |
| Shahraki et al. 2021 [88] | Explored and evaluated Active Learning techniques in Network Traffic Classification | • Demonstrates that Active Learning can achieve high classification accuracy with significantly fewer labeled instances.<br>• Shows how Active Learning techniques can adapt to changes in network traffic, which makes them suitable for dynamic network environments. | • Implementing Active Learning methods can introduce additional computational overhead, particularly in the model retraining phase. | Uncertainty Sampling, Query-By-Committee, and Learning Active Learning |
| Hardegen et al. 2020 [42] | Develop a deep learning model to predict network flow characteristics using real-world network traffic data | • Developed a flow data stream pipeline for real-time training and deployment of deep learning models to predict network flow characteristics.<br>• Collected and analyzed extensive flow data from a university campus network, providing a realistic basis for model training and evaluation. | • The model was trained and validated on data from a university campus network, which may have specific traffic patterns and behaviors. The effectiveness of the model in different network environments remains uncertain.<br>• Other types of ML models, such as RNNs or GNNs, which might be more suited for sequential or structured data, were not explored. | DNNs |

IDS, various unknown data sources in networks can lead to misleading and incorrect results [60]. These attacks can manipulate the model by altering the training data, leading to misclassifying dangerous incidents as safe [15].

Moreover, the dataset commonly utilized in this domain contains information about end-users and providers, which invariably raises significant privacy concerns [49].

The application of ML techniques is noteworthy as an integral component of self-running networks. However, the inherent black-box nature of these algorithms—characterized by a lack of transparency regarding how model decisions are made and outputs generated—raises concerns about the reliability of these networks' decision-making processes. This opacity challenges the trustworthiness of the learning techniques applied therein [49]. One potential solution to mitigate these concerns is adopting the Explainable AI (XAI) methodologies. Nonetheless, the integration of XAI presents substantial challenges due to its developing stage within the broader ML/AI disciplines. In this context, Jacobs et al. [49] conducted research to establish trust in ML models. They proposed the TRUSTEE framework, designed to transform a black-box model—alongside the training dataset—into a white-box model, offering a decision tree-based explanation of the model's reasoning process. However, it is essential to acknowledge that these methods are not devoid of limitations, as they are prone to errors.

### B. Orchestration

While autonomous in individual operations, these advanced networks require a sophisticated orchestration system to ensure that all these elements are integrated and efficiently managed in the networks. For instance, consider a significant fiber link failure. In this scenario, the network's orchestration system must execute multiple tasks. It should quickly reroute traffic to mitigate service disruption (i.e., self-healing) while simultaneously optimizing the rerouted traffic to avoid congestion (i.e., self-optimization). In parallel, it must evaluate whether the failure resulted from a cyber attack, activating necessary self-protection measures. Additionally, the network autonomously configures new routes or devices to manage the load (i.e., self-configuration).

Therefore, this orchestration requires a sophisticated and highly responsive system capable of real-time analysis and decision-making. The system must balance the immediate demands of traffic rerouting and long-term requirements of network efficiency and security. It involves managing multiple autonomous functions in a way that they support and enhance one another, maintaining the network's overall integrity and performance. Hence, the challenge lies in developing an orchestration system to keep pace with the network's self-running nature. It must be intelligent enough to understand and predict the networks' needs and agile enough to respond in real-time to any issues or changes.

### 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 [93].

Because of the heterogeneity of devices, technologies, and protocols within a network, interoperability is one of the main challenges in self-running networks. The solution can lie in universally accepted guidelines to standardize elements in self-running networks. However, providing universal guidelines is not accessible due to the rapid pace of technological evolution in networking. As new technologies emerge, standards that are relevant at one point may quickly become outdated.

Another issue in self-running networks is the lack of uniform data formats and communication protocols, which further complicates the information transition across the network. Hence, this area also required concerted efforts toward standardization.

### D. Ethical Implications and Human Oversight

As networks become more autonomous, leveraging AI and ML to manage, heal, and optimize their functions without human intervention can raise several ethical considerations.

One primary concern is the extent of autonomy granted to self-running networks, especially in critical infrastructures,e.g., healthcare, transportation, and public safety. The delegation of decision-making to algorithms raises questions about the reliability and judgment of these systems in unforeseen situations or emergencies.

Privacy and data protection are also significant concerns. Self-running networks need to process vast amounts of data to learn and make decisions. This capability can clash with the right to privacy if not carefully managed. For this purpose, these networks should make sure that they adhere to only ethical data in the networks. In addition, accountability for decisions made by self-running networks presents a complex challenge. To be more specific, when a network autonomously makes a decision that leads to an adverse outcome, determining responsibility can be difficult. This situation should be defined in a different way than human accountability, so it needs legal and ethical frameworks to establish clear guidelines for oversight, governance, and legal responsibility in the context of self-running networks.

### E. Scalability and Complexity

The integration of diverse vendors, heterogeneous devices, and multifaceted services requires distinct configuration commands to scale networks on demand. Advanced network management techniques can provide significant assistance in addressing these complexities. In response to these challenges, Hong and Zhou [43] introduce NetGraph, a digital twin network platform specifically designed for data center networks. NetGraph generates a virtual replica of the physical network infrastructure, effectively separating configuration data from state data. This separation enhances network management by utilizing IBN, which adds a layer of intelligence to the process. With IBN, operators can specify their desired outcomes (intent), and the system interprets these directives to execute the necessary tasks automatically. The deployment of NetGraph within Huawei's Data Center Networks (DCNs) was undertaken to assess the effectiveness of their model. This approach demonstrated NetGraph's ability to operate in a vendor-agnostic manner, showcasing its versatility and the potential for broader application in various network environments.

### *F. Training and knowledge acquisition*

For training the model, the importance is achieving a balance between speed and accuracy especially for online models. High accuracy often requires extensive computational time, which can conflict with the need for quick processing in real-time applications.

Moreover, obtaining ground truth presents a significant obstacle in the implementation of ML. For supervised learning, ground truth provides the necessary labels for the training process. In the case of unsupervised learning, it is used to assess the model's accuracy. In this regard, two ways are considered: manual labeling and synthetically generating data. While manual annotation allows the incorporation of actual data traces, this method can be time-consuming and prone to errors. On the other hand, Synthetic Generation may result in producing data that may not accurately mirror real-world scenarios [15].

Operating within highly volatile settings, where previously acquired knowledge swiftly becomes outdated, can be challenging for the sake of properly training ML algorithms. However, there are some solutions. For example, Wassermann et al. [99] focus on stream-based ML-based techniques to process each data point individually to ensure that it is analyzed only once and requires minimal memory for operation. These methods are designed to perform efficiently within time limits and offer the notable advantage of allowing predictions to be generated at any juncture in the streaming process.

As we know, training data is the base of using ML algorithms. Nevertheless, the need for huge computational resources for training ML models can raise challenges, which can affect cost, latency, and bandwidth usage [65].

It is worth mentioning that the real or realistic dataset for training the ML models cannot be reached most of the time because this kind of dataset can contain some private traffic patterns, which prevents the industry from exposing data from their users.

## VI. Case Study on Self-Running Networks

This section provides material on how self-running networks can redefine the management and operation of network infrastructures across various domains. By focusing on two case studies on self-running network architecture, i.e., the DCNs and Wireless Sensor Networks (WSNs), we discuss how self-running networks can be used to improve network management. In this regard, the mechanisms and benefits of self-running networks will be elaborated. Through detailed exploration, we reveal how self-running capabilities such as self-configuration, self-optimization, self-protection, and self-healing address the challenges faced by these two case studies, DCNs and WSNs. We found this analysis so helpful since it offers a comprehensive understanding of the potential of self-running networks in supporting the dynamic demands of modern digital ecosystems and the critical role they play in the future of network management.

### *A. Data Center Networks (DCNs)*

The traditional model of DCNs has encountered different challenges, particularly concerning storage, latency, and reliability. These challenges are exacerbated by the increase in data volume, which led to the expansion in network devices within DCNs [56]. By introducing SDN and integrating them into DCNs, the network moved toward automation. However, this level of automation did not address the current complexities and dynamic characteristics of modern networks [62]. Consequently, this situation underscored the need for the next generation of networks that can autonomously manage themselves across all layers. These networks are designed to dynamically reconfigure through changes, optimize their performance, protect against threats, and ensure continuous operation without human intervention.

DCNs must meet specific requirements to support modern digital demands:

- *Low Latency:* Many contemporary applications, especially those requiring real-time interaction, such as cloud computing and video conferencing, depend on low latency for their effective operation. To meet this requirement, DCNs are designed with an emphasis on minimizing transmission delays [104]. However, traditional network architectures are finding it challenging to accommodate the growing demands without a corresponding increase in latency.

- *High Throughput:* To handle the vast amount of data being processed and transferred, DCNs require high throughput. This ensures that data can be moved efficiently across the network without bottlenecks and supports intensive workloads like big data analytics and cloud services [65].

- *Reliability:* DCNs require exceptional reliability to ensure uninterrupted services, especially when faced with unexpected problems. Networks with a design centered on reliability can manage failures smoothly and maintain continuous service without interruption. This requirement would be very important specifically for some applications, such as delay-sensitive applications. In this type of application, the network's ability to swiftly reroute data after a component failure becomes indispensable. However, as networks increase in complexity, the chance of human error escalates, which can potentially compromise network reliability. Moreover, the network should be able to adjust to changes and recover rapidly from setbacks, which underscores the need for proactive problem-solving strategies.

- *Scalability:* As DCNs expand their capacity to ensure that the performance remains unaffected, the ability of network architectures to dynamically adjust resources and efficiently meet changing demands becomes more crucial. Scalability in data center infrastructures makes them keep up with the fast pace of digital transformation and continuous data generation.
- *Security:* Traditional security measures like firewalls, intrusion detection systems, and encryption techniques are increasingly challenged by the sophisticated nature of cyber threats [28]. For example, Advanced Persistent Threats (APTs) and Zero-Day Exploits are complex attacks that can bypass traditional security measures. Hence, security is an essential part of the networking infrastructure that should be considered.

To address these requirements, here is how these self-running capabilities can be employed across various aspects of network management:

**Self-Configuration in DCNs** Whenever we need to reconfigure the system—for instance, when new network devices are added to the data center or when there are changes requiring a reconfiguration—the self-configuration capability comes into play. This function automatically integrates these resources into the network, applying appropriate configurations based on intents and the current state of the network. This process guarantees scalability and low latency and fulfills the requirements of dynamic data center environments. To provide a clear understanding, consider a scenario wherein a data center is tasked with managing an unprecedented volume of data traffic due to an upcoming online event. In conventional DCNs, this situation requires manual intervention to configure additional servers and networking equipment that would be labor-intensive and prone to errors. In contrast, self-running networks with self-configuration capability offer a more efficient solution. Upon deployment of additional hardware, the network instantaneously identifies and assimilates the new resources to have uninterrupted service and peak operational efficiency and eliminates the necessity for manual oversight.

**Self-Optimization in DCNs** Through continuous monitoring and analysis of network performance data, self-optimization functionality can adjust parameters in real time to enhance data flow and reduce bottlenecks, which targets the core requirements of high throughput and low latency. In response to fluctuating workloads and varying application demands, self-optimization algorithms dynamically adjust network resources. This includes optimizing paths for data traffic, allocating bandwidth for critical applications, and balancing loads across servers to have efficient operation and high performance, which results in efficient Resource Allocation. By employing self-optimization, the networks can predict and preemptively adjust to the varying demands placed on the system. For example, during peak usage times, the network can automatically increase bandwidth allocation to critical services or reroute traffic through less congested paths. Reducing Latency through Adaptive Routing and Traffic Management

**Self-Protection in DCNs** Given the paramount importance of security in DCNs, the self-protection capability is crucial. Self-protecting networks leverage advanced algorithms and threat intelligence to identify and mitigate potential security threats in real time. This includes defending against sophisticated cyber attacks such as APTs and Zero-Day Exploits, which traditional security measures may fail to address. By continuously analyzing network traffic for signs of anomalies or malicious activity, self-protecting networks can preemptively block attacks, isolate affected segments, and even patch vulnerabilities before they can be exploited. This proactive approach to network security ensures that the vast volumes of sensitive data stored and processed in data centers remain protected against evolving cyber threats.

**Self-Healing in DCNs** The requirement for reliability is directly satisfied by the self-healing capability of DCNs. Self-healing networks can automatically detect and diagnose failures, whether they are due to hardware malfunctions, software bugs, or external disruptions. Once a problem is identified, the network can initiate corrective actions, such as rerouting traffic or rebooting devices, without human intervention. This ability to swiftly respond to and recover from issues ensures uninterrupted service, a critical requirement for delay-sensitive and mission-critical applications. Moreover, by minimizing the impact of failures and reducing downtime, self-healing networks uphold the principles of fault-tolerant design, further enhancing network reliability. Different research was done to reach this goal. The study by Liu proposes a DRL intelligent routing scheme for software-defined data center networks by using DRL in the control plane to make a routing decision and allocate resources optimally [64]

**1. Sensing Layer** This layer works as a data collector from the physical environment. In the DCNs, this layer is responsible for gathering real-time data from the physical and virtual components throughout the network. Utilizing a variety of sensing devices, this layer captures a wide array of telemetry data essential for network operation and management.

Key telemetry data categories include performance metrics, such as bandwidth usage, latency, packet loss, and throughput, i.e., these are critical for understanding how well the network is performing and for identifying potential bottlenecks or issues affecting data flow and service quality.; network device status, including CPU and memory usage, temperature, and power consumption i.e. high CPU or memory usage, abnormal temperatures, or unusual power consumption can indicate problems that may lead to malfunctions in network devices or network downtime; security metrics, which cover intrusion attempts, malware activity, and traffic anomalies i.e. by monitoring for signs of intrusion, malware, and unusual traffic patterns, the network can react promptly to mitigate risks and protect data and resources; and configuration and system Changes, documented through change logs, i.e., change logs would be useful in auditing, troubleshooting, and understanding the impact of modifications on network performance and security [36].

This layer contains some level of intelligence to do some simple tasks such as data filtering, normalization, or converting the collected data to the proper format for preparing data for transmission to the subsequent layer. Since this layer focuses on data collection, integrating supervised and unsupervised learning for performing these kinds of tasks enhances the layer's capability to process and prepare data for further analysis intelligently.

**2. Data Processing Layer** This layer analyzes the data collected by the sensing layer, processing it to extract meaningful insights and make decisions based on the current network status and predicted trends. The proper intelligence for this layer would be deep learning since deep learning can handle complex pattern recognition and prediction tasks with high accuracy. Moreover, reinforcement learning can be useful because it is suited for making optimization decisions in a dynamic environment, such as adaptive load balancing and resource allocation [64].

**3. Control Layer** Based on the insights and decisions from the data processing layer, the control layer implements changes in the network configuration to optimize performance, ensure reliability, and maintain security. Due to Reinforcement learning's ability to make autonomous decisions that optimize a given performance criterion through trial and error, it will be implemented in this layer. This is particularly effective for real-time network optimization tasks, such as dynamic routing and QoS management.

**4. Application Layer** This layer delivers the network services and applications, ensuring that they meet the performance and reliability requirements of end-users and applications. Supervised learning and deep learning for service personalizing, application performance optimization, and user experience improvements. Supervised learning can be used where historical data and specific outcomes are available, while deep learning can enhance complex decision-making processes and natural language processing for advanced user interfaces.

Applying self-running functionalities in the routing process of DCNs, which is a computing-intensive task, makes DCNs remain reliable and efficient. To be more specific, this process can be enhanced by:

Real-time Traffic Management: Leveraging telemetry data, the DCNs can identify issues as they happen—whether it is an area where a bottleneck happened or a potential security issue threatened. By understanding these issues in real-time, the network can automatically reroute traffic through less congested or more reliable paths to avoid delays and make data safer.

Predictive Routing Adjustments: By analyzing trends and patterns in traffic data, self-running networks can predict future network conditions and proactively adjust routing protocols and configurations to prevent future problems before they happen.

Adaptive Security Measures: Routing decisions also consider security implications, dynamically adapting to mitigate threats as they are detected. For example, traffic can be rerouted away from compromised nodes or through additional security services, i.e., firewalls or intrusion detection systems for inspection. By bringing together real-time traffic management, predictive routing adjustments, and adaptive security measures, self-running capabilities are transforming how the routing process operates.

In this regard, Liu [64] proposed DRL based Routing, a DRL agent deployed on an SDN controller that continually interacts with the network for adaptively performing reasonable routing according to the network state and optimally allocating network resources for traffic.

Lin et al. [63] proposed Primus, a centralized routing mechanism designed for data centers. This method did not suggest any AI/ML solution; instead, it employed centralized controllers for the aggregation of network link state information, subsequently delegating the task of routing computation to the individual switches. The innovation introduced by Primus significantly accelerates the network's convergence time, outperforming both the traditional distributed approach utilized by the Border Gateway Protocol and contemporary centralized routing strategies. Importantly, this advancement is achieved without sacrificing the ability to effectively control and manage routing, suggesting that Primus offers a substantial improvement in both the efficiency and reliability of data center networking operations.

### *B. Wireless Sensor Networks (WSNs)*

WSNs consist of sensor nodes deployed in specific environments where they communicate wirelessly to sense, measure, and gather information. These networks are beneficial across a wide range of applications, such as agriculture, fire hazard monitoring, and underwater exploration. Hence, their environments can be difficult to access, which raises the need for operations that have minimal human intervention. [105]. They have some aspects that should be considered during implementations.

- *Energy Consumption:* WSNs have two main characteristics that make these networks energy-constrained. Firstly, sensors in WSNs are typically powered by batteries or utilize energy-harvesting technologies [78]. As a result, these sensors should minimize energy consumption in order to enhance the QoS and extend the network's lifetime. Moreover, these sensors are often placed in areas with limited access, making it challenging to replenish their energy sources
- *Resilience:* The complex and sensitive environments in which WSNs are deployed, such as in the military, disaster prediction, and healthcare, raise the need for them to be resilient and reliable. These networks should be capable of effectively managing any problem that the network might encounter, e.g., hardware failures and cyber threats, to minimize the disruption due to the challenges and maintain the network functionality [57].
- *Security:* The security challenges have different aspects in WSNs. As the main purpose of Wireless-Sensor Networks (WSN) is monitoring the environment and collecting data, the gathered information, which may contain sensitive data, needs to be protected [7]. Therefore, this network should address security challenges, such as the protection of sensitive data and unauthorized access, to prevent any threats that could compromise data integrity, confidentiality, and network availability. On the other hand, sensor

nodes are accessible and interact with people, which can also pose security challenges [57].

Here is how these self-running capabilities can address WSNs challenges:

**Self-Configuration for WSNs** This functionality enables sensors to automatically establish and adjust their connections with neighboring nodes when deployed in a new environment. Upon ongoing changes, the self-configuration process adapts the network to the current situation based on the current network status and environmental conditions. This process can include several things, such as establishing an optimal routing topology without operator intervention and reorganizing the network to have a steady connection and proper network performance. Optimizing network operations enables the system to operate with minimal energy consumption [103]. This intelligent energy management prolongs the lifespan of the network and allows it to operate at peak efficiency under varying conditions.

**Self-Protection for WSNs** It implements security measures that allow the network to detect and mitigate threats autonomously. These threats can have different natures. For instance, if WSN is deployed in a remote forest area for fire hazard monitoring, the potential threat the network faces would be malicious attacks, data interception during data transmission, or even physical tampering by wildlife. Self-protection techniques can include intrusion detection systems to monitor the network traffic for recognizing any malicious patterns or anomalies [57].

Moreover, as WSNs have a continuous change in their topology, limited resources, and the absence of centralized control, self-protection mechanisms ensure these networks work properly even when faced with technical, malicious, or environmental challenges [5].

**Self-Optimization for WSNs** This functionality is great where energy resources are often limited. Through the implementation of various adaptive algorithms and models, self-optimization enables these networks to adjust their operations, such as data transmission rates, sensor activation schedules, and routing protocols, in response to environmental changes or varying network conditions. This approach aims to minimize energy consumption while maintaining network performance. In addition to that, by dynamically modifying their behavior, WSNs can enhance the network lifespan and resilience. This improvement makes them more suitable for a wide range of applications, from environmental monitoring to smart cities. Furthermore, many of these applications are delay-sensitive, so by leveraging self-optimization capability, the networks can reduce the latency in WSNs.

**Self-Healing for WSNs** Self-healing mechanisms enhance the resilience of WSNs in this untrusted, complicated environment that has energy and computational resources challenges [5]. The robustness of the network against various challenges can be guaranteed by applying self-healing because this mechanism empowers WSNs to detect issues, make decisions, and recover from faults without human intervention. This resilience makes self-healing capability a fundamental aspect of WSNs design and deployment that addresses the networks' vulnerability and ensures their effectiveness in fulfilling their intended roles.

There are other problems in this type of network, too, such as the coverage problem that wants to have minimal overlap between sensors[103]. By using AI, there is a way to find the optimal locations for sensor placement. However, the routing problem, i.e., finding the best way to transfer data to ensure data accuracy and minimize energy consumption, is one of the most important problems that can affect networks' energy consumption, resilience, and security [103]. In WSNs, routing significantly influences energy consumption and the lifespan of the network [87]. Traditional approaches to routing in WSNs have faced challenges in simultaneously managing various critical aspects. These aspects include ensuring all sensors in a specific area are aware of their location, designing and applying solutions across different fields, prioritizing energy efficiency to maximize network life (considering the difficulty in accessing many sensor nodes), and incorporating QoS to meet user expectations for service quality. Addressing these challenges is crucial for developing effective routing strategies in WSNs [87]. In this way, the goal is to reach the routing mechanism that would be secure and energy efficient [80] [26]. For example, the study [11]uses network pruning to find the proper route in WSNs.

The intelligence required at various layers of self-running networks in WSNs can be described as follows. **1. Sensing Layer** The sensing layer is typically the most resource-constrained. The sensors can implement lightweight ML-based models [11]. By analyzing network traffic for network management, AI algorithms can aggregate and do data filtering and compression [57]. This leads to reducing the volume of data that needs to be transmitted. Supervised learning for traffic analysis, such as Decision trees or linear regression, can be implemented in this layer. Also, techniques that reduce data dimensionality, e.g., PCA for feature reduction or algorithms for selecting the most informative features, can help minimize the data that needs to be transmitted to upper layers.

This layer is responsible for collecting information on the in-network state and telemetry data. Telemetry data in WSN can be categorized based on the type of information collected, such as environmental data, i.e., temperature, humidity, air pressure, light levels, and pollution metrics. This type of telemetry data is common in environmental monitoring, agriculture, and weather forecasting. Motion and location data, i.e., position, velocity, and orientation data, are collected using GPS sensors, accelerometers, and gyroscopes. This data is crucial for applications in logistics, wildlife tracking, and smart transportation systems. Health and biometric data, i.e., heart rate, body temperature, and other physiological parameters are monitored in health care and fitness applications. Structural Data, i.e., strain, pressure, and vibrations, were measured on buildings, bridges, and other structures to assess their integrity and safety.

**2. Data Processing Layer** This layer can leverage both labeled and unlabeled data. Techniques like clustering, e.g., k-means, can aggregate data from various sensors to reduce redundancy before further processing or transmission.

Moreover, for more complex processing that cannot be handled by the sensing layer but does not require central processing, deploying neural network models optimized for edge computing can enable real-time decision-making directly on gateway devices.

**3. Control Layer** This layer is responsible for various complex tasks such as load balancing and routing. Hence, one of the best intelligence for this kind of task would be reinforcement learning [57]. reinforcement learning can be effective for dynamic decision-making and optimization of network operations since reinforcement learning agents can learn optimal policies for various operations by interacting with the environment.

**4. Application Layer** For applications requiring complex data analysis, prediction, or pattern recognition, such as image or sound analysis, deep learning techniques such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) can be used. Also, for applications involving human interaction, NLP techniques can enable the extraction of actionable insights from textual data or provide natural language interfaces.

## VII. Conclusion

Self-running networks can play a powerful role in network management. It utilizes the intelligence in each part of its architecture that enables it to handle the dynamic and complex nature of modern networks. In this paper, we explored the self-running networks from the components and functionality to previous research towards them. We proposed the unified architecture of four self-running networks and how each layer of the architecture should be. We concluded our work by exploring these networks for two common network types, DCN and WSN, and showing how self-running networks can satisfy their requirements.

## Appendix A
## List of Abbreviations

**SDN** Software-Defined Networking
**NFV** Network Functions Virtualization
**SDR** Software-Defined Radio
**IBN** Intent-Based Networking
**WSN** Wireless-Sensor Networks
**IoT** Internet of Things
**ML** Machine Learning
**NLP** Natural Language Processing
**QoS** Quality of Service
**QoE** Quality of Experience
**AI** Artificial Intelligence
**GPUs** Graphics Processing Units
**TPUs** Tensor Processing Units
**API** Application Programming Interface
**5G** 5th Generation Mobile Network
**ETSI** European Telecommunications Standards Institute
**MDs** Management Domains
**ZSM** Zero-touch network and Service Management
**IETF** Internet Engineering Task Force
**PBNM** Policy-Based Network Management
**NMS** Network Management System
**DDoS** Distributed Denial of Service
**HAN** Home Area Network
**VoIP** Voice over Internet Protocol
**SAE** Society of Automotive Engineers
**ITU** International Telecommunication Union
**OPEX** OPerating EXpenses
**IDS** Intrusion Detection Systems
**XAI** Explainable AI
**ANM** Autonomic Networks Management
**RAN** Radio Access Network
**DCNs** Data Center Networks
**WSNs** Wireless Sensor Networks
**DRL** Deep Reinforcement Learning
**IDN** Intent-driven networks
**AGRU** Attention-based Gated Recurrent Units
**APTs** Advanced Persistent Threats
**CNNs** Convolutional Neural Networks
**DNNs** Deep Neural Networks
**RNNs** Recurrent Neural Networks

## References

[1] June 15 and Stephen Watts. An introduction to self-driving networks, Jun 2020.

[2] Khizar Abbas, Muhammad Afaq, Talha Ahmed Khan, Adeel Rafiq, and Wang-Cheol Song. Slicing the core network and radio access network domains through intent-based networking for 5g networks. *Electronics*, 9(10):1710, 2020.

[3] Khizar Abbas, Talha Ahmed Khan, Muhammad Afaq, and Wang-Cheol Song. Network slice lifecycle management for 5g mobile networks: An intent-based networking approach. *IEEE Access*, 9:80128–80146, 2021.

[4] Khizar Abbas, Talha Ahmed Khan, Muhammad Afaq, and Wang-Cheol 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*, pages 1–5. IEEE, 2022.

[5] Rami Ahmad, Raniyah Wazirali, and Tarik Abu-Ain. Machine learning for wireless sensor networks security: An overview of challenges and issues. *Sensors*, 22(13):4730, 2022.

[6] Mustafa K Mehmet Ali and Faouzi Kamoun. Neural networks for shortest path computation and routing in computer networks. *IEEE transactions on neural networks*, 4(6):941–954, 1993.

[7] Mohammed Aljebreen, Manal Abdullah Alohali, Muhammad Kashif Saeed, Heba Mohsen, Mesfer Al Duhayyim, Amgad Atta Abdelmageed, Suhanda Drar, and Sitelbanat Abdelbagi. Binary chimp optimization algorithm with ml based intrusion detection for secure iot-assisted wireless sensor networks. *Sensors*, 23(8):4073, 2023.

[8] Mhd Saria Allahham, Alaa Awad Abdellatif, Naram Mhaisen, Amr Mohamed, Aiman Erbad, and Mohsen Guizani. Multi-agent reinforcement learning for network selection and resource allocation in heterogeneous multi-rat networks. *IEEE Transactions on Cognitive Communications and Networking*, 8(2):1287–1300, 2022.

[9] Hani Alquhayz, Nasser Alalwan, Ahmed Ibrahim Alzahrani, Ali H Al-Bayatti, and Mhd Saeed Sharif. Policy-based security management system for 5g heterogeneous networks. *Wireless Communications and Mobile Computing*, 2019:1–14, 2019.

[10] Jimena Andrade-Hoz, Qi Wang, and Jose M Alcaraz-Calero. Infrastructure-wide and intent-based networking dataset for 5g-and-beyond ai-driven autonomous networks. *Sensors*, 24(3):783, 2024.

[11] Davide Andreoletti, Cristina Rottondi, Fatima Ezzeddine, Omran Ayoub, and Silvia Giordano. Ml-based network pruning for routing data overhead reduction in wireless sensor networks. In *2023 18th Wireless On-Demand Network Systems and Services Conference (WONS)*, pages 122–125. IEEE, 2023.

[12] Bini Angui, Romuald Corbel, Veronica Quintuna Rodriguez, and Emile Stephan. Towards 6g zero touch networks: The case of automated cloud-ran deployments. In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, 2022.

[13] Sisay Tadesse Arzo, Riccardo Bassoli, Fabrizio Granelli, and Frank HP Fitzek. Multi-agent based autonomic network management architecture. *IEEE Transactions on Network and Service Management*, 18(3):3595–3618, 2021.

[14] Sisay Tadesse Arzo, Claire Naiga, Fabrizio Granelli, Riccardo Bassoli, Michael Devetsikiotis, and Frank HP Fitzek. A theoretical discussion and survey of network automation for iot: Challenges and opportunity. *IEEE Internet of Things Journal*, 8(15):12021–12045, 2021.

[15] Sara Ayoubi, Noura Limam, Mohammad A Salahuddin, Nashid Shahriar, Raouf Boutaba, Felipe Estrada-Solano, and Oscar M Caicedo. Machine learning for cognitive network management. *IEEE Communications Magazine*, 56(1):158–165, 2018.

[16] Dario Bega, Marco Gramaglia, Marco Fiore, Albert Banchs, and Xavier Costa-Perez. Deepcog: Cognitive network management in sliced 5g networks with deep learning. In *IEEE INFOCOM 2019-IEEE conference on computer communications*, pages 280–288. IEEE, 2019.

[17] Chafika Benzaid and Tarik Taleb. Ai-driven zero touch network and service management in 5g and beyond: Challenges and research directions. *Ieee Network*, 34(2):186–194, 2020.

[18] Wafa Berrayana, Habib Youssef, and Guy 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)*, pages 82–87. IEEE, 2012.

[19] Yi-Ren Chen, Amir Rezapour, Wen-Guey Tzeng, and Shi-Chun Tsai. Rl-routing: An sdn routing algorithm based on deep reinforcement learning. *IEEE Transactions on Network Science and Engineering*, 7(4):3185–3199, 2020.

[20] Yu Cheng, Ramy Farha, Myung Sup Kim, Alberto Leon-Garcia, and James Won-Ki Hong. A generic architecture for autonomic service and network management. *Computer Communications*, 29(18):3691–3709, 2006.

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

[22] Alan Collet, Albert Banchs, and Marco Fiore. Lossleap: Learning to predict for intent-based networking. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pages 2138–2147. IEEE, 2022.

[23] Estefania Coronado, Rasoul Behravesh, Tejas Subramanya, Adriana Fernández-Fernández, Shuaib Siddiqui, Xavier Costa-Pérez, and Roberto Riggio. Zero touch management: A survey of network automation solutions for 5g and 6g networks. *IEEE Communications Surveys & Tutorials*, 2022.

[24] Ed. J. Boyle R. Cohen S. Herzog R. Rajan A. Sastry D. Durham. The cops (common open policy service) protocol, 2000.

[25] Nathan F Saraiva de Sousa, Danny Lachos Perez, Christian Esteve Rothenberg, and Pedro Henrique Gomes. End-to-end service monitoring for zero-touch networks. *Journal of ICT Standardization*, 9(2):91–112, 2021.

[26] Bakkiam David Deebak and Fadi Al-Turjman. A hybrid secure routing and monitoring mechanism in iot-based wireless sensor networks. *Ad Hoc Networks*, 97:102022, 2020.

[27] Hajer Derbel, Nazim Agoulmine, and Mikaël Salaün. Anema: Autonomic network management architecture to support self-configuration and self-optimization in ip networks. *Computer Networks*, 53(3):418–430, 2009.

[28] Haiwei Dong, Ali Munir, Hanine Tout, and Yashar Ganjali. Next-generation data center network enabled by machine learning: Review, challenges, and opportunities. *IEEE Access*, 9:136459–136475, 2021.

[29] Chip Elliott and Bob Heile. Self-organizing, self-healing wireless networks. In *2000 IEEE Aerospace Conference. Proceedings (Cat. No. 00TH8484)*, volume 1, pages 149–156. IEEE, 2000.

[30] ETSI. Zero-touch network and service management (zsm); means of automation. https://www.etsi.org/technologies/zero-touch-network-service-management, 2020. Accessed: 2017.

[31] Zubair Md Fadlullah, Fengxiao Tang, Bomin Mao, Nei Kato, Osamu Akashi, Takeru Inoue, and Kimihiro Mizutani. State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. *IEEE Communications Surveys & Tutorials*, 19(4):2432–2455, 2017.

[32] Liam Fallon, John Keeney, and Ram Krishna Verma. Autonomic closed control loops for management, an idea whose time has come? In *2019 15th International Conference on Network and Service Management (CNSM)*, pages 1–5. IEEE, 2019.

[33] Jeroen Famaey, Steven Latré, John Strassner, and Filip De Turck. A hierarchical approach to autonomic network management. In *2010 IEEE/IFIP Network Operations and Management Symposium Workshops*, pages 225–232. Ieee, 2010.

[34] Nick Feamster and Jennifer Rexford. Why (and how) networks should run themselves. *arXiv preprint arXiv:1710.11583*, 2017.

[35] Nick Feamster, Jennifer Rexford, and Ellen Zegura. The road to sdn: an intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review*, 44(2):87–98, 2014.

[36] Shir Landau Feibish, Zaoxing Liu, and Jennifer Rexford. Compact data structures for network telemetry. *arXiv preprint arXiv:2311.02636*, 2023.

[37] Diogo Ferreira, Andre Braga Reis, Carlos Senna, and Susana Sargento. A forecasting approach to improve control and management for 5g networks. *IEEE Transactions on Network and Service Management*, 18(2):1817–1831, 2021.

[38] Jorge Gallego-Madrid, Ramon Sanchez-Iborra, Pedro M Ruiz, and Antonio F Skarmeta. Machine learning-based zero-touch network and service management: A survey. *Digital Communications and Networks*, 8(2):105–123, 2022.

[39] Alan G Ganek and Thomas A Corbi. The dawning of the autonomic computing era. *IBM systems Journal*, 42(1):5–18, 2003.

[40] R. Yavatkar D. Pendarakis R. Guerin. A framework for policy-based admission control, 2000.

[41] Sylvain Hallé, Éric Wenaas, Roger Villemaire, and Omar Cherkaoui. Self-configuration of network devices with configuration logic. In *IFIP TC6 International Conference on Autonomic Networking*, pages 36–49. Springer, 2006.

[42] Christoph Hardegen, Benedikt Pfülb, Sebastian Rieger, and Alexander Gepperth. Predicting network flow characteristics using deep learning and real-world network traffic. *IEEE Transactions on Network and Service Management*, 17(4):2662–2676, 2020.

[43] Hanshu Hong, Qin Wu, Feng Dong, Wei Song, Ronghua Sun, Tao Han, Cheng Zhou, and Hongwei Yang. Netgraph: An intelligent operated digital twin platform for data center networks. In *Proceedings of the ACM SIGCOMM 2021 workshop on network-application integration*, pages 26–32, 2021.

[44] Huakun Huang, Lingjun Zhao, Huawei Huang, and Song Guo. Machine fault detection for intelligent self-driving networks. *IEEE Communications Magazine*, 58(1):40–46, 2020.

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

[46] Markus C Huebscher and Julie A McCann. A survey of autonomic computing—degrees, models, and applications. *ACM Computing Surveys (CSUR)*, 40(3):1–28, 2008.

[47] IEEE. The Levels of Intelligence of Mobile Networks and Consideration of Architecture Evolution. https://wwwfuturenetworks.ieee.org/tech-focus/december-2018/levels-of-intelligence, year=2018.

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

[49] Arthur S Jacobs, Roman Beltiukov, Walter Willinger, Ronaldo A Ferreira, Arpit Gupta, and Lisandro 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*, pages 1537–1551, 2022.

[50] Arthur S Jacobs, Ricardo J Pfitscher, Rafael H Ribeiro, Ronaldo A Ferreira, Lisandro Z Granville, Walter Willinger, and Sanjay G Rao. Hey, lumi! using natural language for {intent-based} network management. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 625–639, 2021.

[51] Arthur Selle Jacobs, Ronaldo Alves Ferreira, and Lisandro Zambenetti Granville. Enabling self-driving networks with machine learning. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6. IEEE, 2023.

[52] Arthur Selle Jacobs, Ricardo José Pfitscher, Ronaldo Alves Ferreira, and Lisandro Zambenedetti Granville. Refining network intents for self-driving networks. In *Proceedings of the Afternoon Workshop on Self-Driving Networks*, pages 15–21, 2018.

[53] Brendan Jennings, Sven Van Der Meer, Sasitharan Balasubramaniam, Dmitri Botvich, Mícheál Ó Foghlú, William Donnelly, and John Strassner. Towards autonomic management of communications networks. *IEEE Communications Magazine*, 45(10):112–121, 2007.

[54] Wei Jiang, Mathias Strufe, and Hans Schotten. Autonomic network management for software-defined and virtualized 5g systems. In

*European Wireless 2017; 23th European Wireless Conference*, pages 1–6. VDE, 2017.

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

[56] Juniper Networks. What is a data center network? https://www.juniper.net/us/en/research-topics/what-is-a-data-center-network.html, 2024. Accessed: 2024-03-27.

[57] F Fernando Jurado-Lasso, Letizia Marchegiani, Jesus Fabian Jurado, Adnan M Abu-Mahfouz, and Xenofon Fafoutis. A survey on machine learning software-defined wireless sensor networks (ml-sdwsns): Current status and major challenges. *IEEE Access*, 10:23560–23592, 2022.

[58] Patrick Kalmbach, Johannes Zerwas, Péter Babarczi, Andreas Blenk, Wolfgang Kellerer, and Stefan Schmid. Empowering self-driving networks. In *Proceedings of the afternoon workshop on self-driving networks*, pages 8–14, 2018.

[59] Jeffrey O Kephart and David M Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[60] Randhir Kumar, Prabhat Kumar, Moayad Aloqaily, and Ahamed Aljuhani. Deep-learning-based blockchain for secure zero touch networks. *IEEE Communications Magazine*, 61(2):96–102, 2022.

[61] Aris Leivadeas and Matthias Falkner. A survey on intent based networking. *IEEE Communications Surveys & Tutorials*, 2022.

[62] Bo Li, Ting Wang, Peng Yang, Mingsong Chen, Shui Yu, and Mounir Hamdi. Machine learning empowered intelligent data center networking: A survey. *arXiv preprint arXiv:2202.13549*, 2022.

[63] Fusheng Lin, Hongyu Wang, Guo Chen, Guihua Zhou, Tingting Xu, Dehui Wei, Li Chen, Yuanwei Lu, Andrew Qu, Hua Shao, et al. Fast, scalable and robust centralized routing for data center networks. *IEEE/ACM Transactions on Networking*, 2023.

[64] Wai-xi Liu. Intelligent routing based on deep reinforcement learning in software-defined data-center networks. In *2019 IEEE symposium on computers and communications (ISCC)*, pages 1–6. IEEE, 2019.

[65] Yi Liu, Jiangping Han, Kaiping Xue, Jian Li, Qibin Sun, and Jun Lu. Decc: Achieving low latency in data center networks with deep reinforcement learning. *IEEE Transactions on Network and Service Management*, 2023.

[66] Madhusanka Liyanage, Quoc-Viet Pham, Kapal Dev, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu, and Gokul Yenduri. A survey on zero touch network and service management (zsm) for 5g and beyond networks. *Journal of Network and Computer Applications*, 203:103362, 2022.

[67] Xinjian Long, Xiangyang Gong, Xirong Que, Wendong Wang, Bing Liu, Sheng Jiang, and Ning Kong. Autonomic networking: Architecture design and standardization. *IEEE Internet Computing*, 21(5):48–53, 2017.

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

[69] Tianle Mai, Sahil Garg, Haipeng Yao, Jiangtian Nie, Georges Kaddoum, and Zehui Xiong. In-network intelligence control: Toward a self-driving networking architecture. *IEEE Network*, 35(2):53–59, 2021.

[70] Barbara Martini, Molka Gharbaoui, and Piero Castoldi. Intent-based network slicing for sdn vertical services with assurance: Context, design and preliminary experiments. *Future Generation Computer Systems*, 142:101–116, 2023.

[71] Federico Mason, Gianfranco Nencioni, and Andrea Zanella. Using distributed reinforcement learning for resource orchestration in a network slicing scenario. *IEEE/ACM Transactions on Networking*, 31(1):88–102, 2023.

[72] Joseph Mcnamara, Daniel Camps-Mur, Meysam Goodarzi, Hilary Frank, Lorena Chinchilla-Romero, Ferrán Cañellas, Adriana Fernández-Fernández, and Shuangyi Yan. Nlp powered intent based network management for private 5g networks. *IEEE Access*, 11:36642–36657, 2023.

[73] Kashif Mehmood, HV Kalpanie Mendis, Katina Kralevska, and Poul E Heegaard. Intent-based network management and orchestration for smart distribution grids. In *2021 28th International Conference on Telecommunications (ICT)*, pages 1–6. IEEE, 2021.

[74] Abdelkader Mekrache, Adlen Ksentini, and Christos Verikoukis. Intent-based management of next-generation networks: an llm-centric approach. *IEEE Network*, 2024.

[75] Jessica Moysen and Lorenza Giupponi. From 4g to 5g: Self-organized network management meets machine learning. *Computer Communications*, 129:248–268, 2018.

[76] Bruno Astuto A Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications surveys & tutorials*, 16(3):1617–1634, 2014.

[77] Barbara Orlandi, Sandrine Lataste, Sylvaine Kerboeuf, Marc Bouillon, Xiaofeng Huang, Frédéric Faucheux, Arzhang Shahbazi, and Pascal 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)*, pages 163–170. IEEE, 2024.

[78] Walid Osamy, Ahmed M Khedr, Ahmed Salim, Amal Ibrahim Al Ali, and Ahmed A El-Sawy. Coverage, deployment and localization challenges in wireless sensor networks based on artificial intelligence techniques: a review. *IEEE Access*, 10:30232–30257, 2022.

[79] Fannia Pacheco, Ernesto Exposito, Mathieu Gineste, Cedric Baudoin, and Jose Aguilar. Towards the deployment of machine learning solutions in network traffic classification: A systematic survey. *IEEE Communications Surveys & Tutorials*, 21(2):1988–2014, 2018.

[80] Rahul Priyadarshi. Energy-efficient routing in wireless sensor networks: A meta-heuristic and artificial intelligence-based approach: A comprehensive review. *Archives of Computational Methods in Engineering*, pages 1–29, 2024.

[81] Adeel Rafiq, Muhammad Afaq, and Wang-Cheol Song. Intent-based networking with proactive load distribution in data center using ibn manager and smart path manager. *Journal of Ambient Intelligence and Humanized Computing*, 11:4855–4872, 2020.

[82] Annie Ibrahim Rana and Mícheál Ó Foghlú. Policy-based network management in home area networks: Interim test results. In *2009 3rd International Conference on New Technologies, Mobility and Security*, pages 1–3. IEEE, 2009.

[83] Farhad Rezazadeh, Hatim Chergui, Luis Alonso, and Christos Verikoukis. Continuous multi-objective zero-touch network slicing via twin delayed ddpg and openai gym. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020.

[84] Mohammad Riftadi and Fernando Kuipers. P4i/o: Intent-based networking with p4. In *2019 IEEE Conference on Network Softwarization (NetSoft)*, pages 438–443. IEEE, 2019.

[85] Dario Rossi and Liang Zhang. Landing ai on networks: An equipment vendor viewpoint on autonomous driving networks. *IEEE Transactions on Network and Service Management*, 19(3):3670–3684, 2022.

[86] SAE International. Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems. https://www.sae.org/standards/content/j3016_201401/preview/, 2014.

[87] Amit Sarkar and T Senthil Murugan. Routing protocols for wireless sensor networks: What the literature says? *Alexandria Engineering Journal*, 55(4):3173–3183, 2016.

[88] Amin Shahraki, Mahmoud Abbasi, Amir Taherkordi, and Anca Delia Jurcut. Active learning for network traffic classification: a technical study. *IEEE Transactions on Cognitive Communications and Networking*, 8(1):422–439, 2021.

[89] Amritpal Singh, Gagangeet Singh Aujla, and Rasmeet Singh Bali. Intent-based network for data dissemination in software-defined vehicular edge computing. *IEEE Transactions on Intelligent Transportation Systems*, 22(8):5310–5318, 2020.

[90] Morris Sloman. Policy driven management for distributed systems. *Journal of network and Systems Management*, 2:333–360, 1994.

[91] Thato Solomon, Adamu Murtala Zungeru, Rajalakshmi Selvaraj, Olefile Phakedi, and Ontiretse Bagwasi. Policy-based network management in biust network. *American Journal of Engineering and Applied Sciences*, 10(3):661–668, Jun 2017.

[92] Adamantia Stamou, Nikos Dimitriou, Kimon Kontovasilis, and Symeon Papavassiliou. Autonomic handover management for heterogeneous networks in a future internet context: A survey. *IEEE Communications Surveys & Tutorials*, 21(4):3274–3297, 2019.

[93] Yaohua Sun, Mugen Peng, Yangcheng Zhou, Yuzhe Huang, and Shiwen Mao. Application of machine learning in wireless networks: Key techniques and open issues. *IEEE Communications Surveys & Tutorials*, 21(4):3072–3108, 2019.

[94] Fengxiao Tang, Bomin Mao, Yuichi Kawamoto, and Nei Kato. Survey on machine learning for intelligent end-to-end communication toward 6g: From network access, routing to traffic control and streaming adaption. *IEEE Communications Surveys & Tutorials*, 23(3):1578–1598, 2021.

[95] Kostas Tsagkaris, Marios Logothetis, Vassilis Foteinos, George Poulios, Michalis Michaloliakos, and Panagiotis Demestichas. Customizable autonomic network management: integrating autonomic network management and software-defined networking. *IEEE Vehicular Technology Magazine*, 10(1):61–68, 2015.

[96] Benjamin E Ujcich, Adam Bates, and William H Sanders. Provenance for intent-based networking. In *2020 6th IEEE conference on network softwarization (NetSoft)*, pages 195–199. IEEE, 2020.

[97] Muhammad Usama, Junaid Qadir, Aunn Raza, Hunain Arif, Kok-Lim Alvin Yau, Yehia Elkhatib, Amir Hussain, and Ala Al-Fuqaha. Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE access*, 7:65579–65615, 2019.

[98] Dinesh C Verma. Simplifying network administration using policy-based management. *IEEE network*, 16(2):20–26, 2002.

[99] Sarah Wassermann, Thibaut Cuvelier, Pavol Mulinka, and Pedro Casas. Adaptive and reinforcement learning approaches for online network monitoring and analysis. *IEEE Transactions on Network and Service Management*, 18(2):1832–1849, 2020.

[100] Shalitha Wijethilaka and Madhusanka Liyanage. Survey on network slicing for internet of things realization in 5g networks. *IEEE Communications Surveys & Tutorials*, 23(2):957–994, 2021.

[101] Bruno Missi Xavier, Rafael Silva Guimarães, Giovanni Comarela, and Magnos Martinello. Map4: A pragmatic framework for in-network machine learning traffic classification. *IEEE Transactions on Network and Service Management*, 19(4):4176–4188, 2022.

[102] Helin Yang, Arokiaswami Alphones, Zehui Xiong, Dusit Niyato, Jun Zhao, and Kaishun Wu. Artificial-intelligence-enabled intelligent 6g networks. *IEEE Network*, 34(6):272–280, 2020.

[103] Jian Yang and Yimin Xia. Coverage and routing optimization of wireless sensor networks using improved cuckoo algorithm. *IEEE Access*, 2024.

[104] Zhiyuan Yao, Yoann Desmouceaux, Mark Townsley, and Thomas Heide Clausen. Towards intelligent load balancing in data centers. *arXiv preprint arXiv:2110.15788*, 2021.

[105] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.

[106] Shuai Yu, Xu Chen, Zhi Zhou, Xiaowen Gong, and Di 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*, 8(4):2238–2251, 2020.

[107] Noe M Yungaicela-Naula, Cesar Vargas-Rosales, Jesús Arturo Pérez-Díaz, and Mahdi Zareei. Towards security automation in software defined networks. *Computer Communications*, 183:64–82, 2022.

[108] Alessio Zappone, Marco Di Renzo, and Mérouane Debbah. Wireless networks design in the era of deep learning: Model-based, ai-based, or both? *IEEE Transactions on Communications*, 67(10):7331–7376, 2019.

[109] Engin Zeydan and Yekta Turk. Recent advances in intent-based networking: A survey. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5. IEEE, 2020.

[110] Haijun Zhang, Haisen Zhang, Keping Long, and George K Karagiannidis. Deep learning based radio resource management in noma networks: User association, subchannel and power allocation. *IEEE Transactions on Network Science and Engineering*, 7(4):2406–2415, 2020.

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

[112] Xiaoang Zheng, Aris Leivadeas, and Matthias Falkner. Intent based networking management with conflict detection and policy resolution in an enterprise network. *Computer Networks*, 219:109457, 2022.