ISCTE IUL
REPOSITÓRIO
INSTITUTO UNIVERSITÁRIO DE LISBOA

Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:
2026-10-28

Deposited version:
Accepted Version

Peer-review status of attached file:
Peer-reviewed

Citation for published item:
Moura, J. & Santana, P. (2025). Toward end-to-end deep learning for autonomous management in next-generation networks. 2025 Sixteenth International Conference on Ubiquitous and Future Networks (ICUFN) (pp. 181?186). IEEE. https://doi.org/10.1109/ICUFN65838.2025

Further information on publisher's website:
10.1109/ICUFN65838.2025

Publisher's copyright statement:
This is the peer reviewed version of the following article: Moura, J. & Santana, P. (2025). Toward end-to-end deep learning for autonomous management in next-generation networks. 2025 Sixteenth International Conference on Ubiquitous and Future Networks (ICUFN) (pp. 181?186). IEEE. https://doi.org/10.1109/ICUFN65838.2025, which has been published in final form at https://dx.doi.org/10.1109/ICUFN65838.2025. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0
The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Serviços de Informação e Documentação, Instituto Universitário de Lisboa (ISCTE-IUL)
Av. das Forças Armadas, Edifício II, 1649-026 Lisboa Portugal
Phone: +(351) 217 903 024 | e-mail: administrador.repositorio@iscte-iul.pt
https://repositorio.iscte-iul.pt

Toward End-to-End Deep Learning for Autonomous Management in Next-Generation Networks

José Moura[1,2], Pedro Santana[1,3]
[1]ISCTE—University Institute of Lisbon, Lisbon, Portugal
[2]Instituto de Telecomunicações, Lisbon, Portugal
[3]ISTAR—Information Sciences and Technologies and Architecture Research Center, Lisbon, Portugal
Email: jose.moura@iscte-iul.pt, pedro.santana@iscte-iul.pt

***Abstract*—The evolution towards next-generation of mobile networks demands for autonomous network management, emphasizing data-driven solutions based on Artificial Intelligence (AI), in particular machine learning. To attain such goals, this paper proposes a hybrid end-to-end learning approach that integrates imitation learning, deep reinforcement learning, simulation, domain adaptation, multi-agent cooperation, explainable AI, and generative AI. The work outlines a comprehensive vision for online agent learning about optimum network management policies while ensuring safety, interpretability, and adaptability in highly complex and dynamic use cases at the network periphery.**

***Index Terms*—Programmable Networks, Reinforcement & Imitation Learning, End-to-End Learning, Network Automation, Transfer Learning, Safe RL, Generative AI, Explainable AI**

I. Introduction

Network administration remains one of the most complex, error-prone, and labor-intensive aspects of computer networking. Next-Generation mobile network environments face unprecedented operational challenges [1]–[5]. These include performance tuning, fault diagnosis, policy enforcement, and dynamic reconfiguration.

Software-Defined Networking (SDN) provides a clean separation between the control and data planes and exposes programmable interfaces for configuring network behavior [6]. However, current SDN controllers still rely on static policies or manual operator inputs for key decisions. This motivates developing intelligent agents that can reason and (semi-)autonomously manage SDN-controlled network resources.

In this context, the paper proposes to tackle autonomous management of mobile edge networks with an AI Toolbox based on end-to-end deep learning solutions. A visual representation of the main elements of this toolbox is given in Fig. 1. We build upon prior surveys that highlight the benefits of applying machine learning techniques to computer networks [7]–[10]. We also provide a comprehensive discussion of the employed AI technologies, highlighting their individual contributions to enabling end-to-end autonomous agent learning. These agents can continuously adapt and derive optimal network management policies in response to changing and unexpected network statuses.

The remaining paper's structure is as follows. Section II debates the SDN's role in enabling intelligent and flexible network administration. Section III outlines a learning-centric roadmap for achieving autonomous management at the programmable network edge. In Section IV, we examine some of the practical obstacles for applying the AI toolbox in real-world programmable edge network environments. Finally, Section V finalizes the paper and points out next promising research directions to the deployment of the AI toolbox on edge networking environments.

II. SDN in Network Administration

A. Key SDN Features for Learning Agents

SDN [6] offers several characteristics that are highly amenable to learning-based control. First, its centralized control architecture provides the SDN's controller with global awareness of the network's topology, traffic matrices, and the state of all connected devices. Second, its programmable interfaces enable the controller to dynamically modify forwarding tables, reroute flows, or block traffic using protocols such as OpenFlow. Third, SDN systems expose a rich set of internal state information, including low-level statistics like packet-in events, flow counters, and port statistics, which can serve as valuable input features for learning agents. Finally, the SDN controllers integrate well with learning-based agents as independent applications, facilitating experimentation and deployment without disrupting core network functionality.

B. Mapping Learning to SDN Control Tasks

We categorize several key SDN tasks where learning agents can be effectively applied. One major area is flow management, where agents select and install flow rules based on observed traffic patterns and application requirements; this encompasses both reactive strategies, where rules are installed on demand, and proactive strategies, where rules are pre-installed in anticipation of traffic. Another critical task is traffic engineering, in which agents learn to balance loads across multiple paths, reroute flows to avoid congestion, and prioritize latency-sensitive traffic. In topology control, learning agents can manage link activations, detect failures, and reconfigure routes in real time. For policy enforcement, agents can be trained to identify traffic that violates defined network policies and to apply corrective measures automatically. Lastly, in the domain of fault detection and recovery, agents can detect

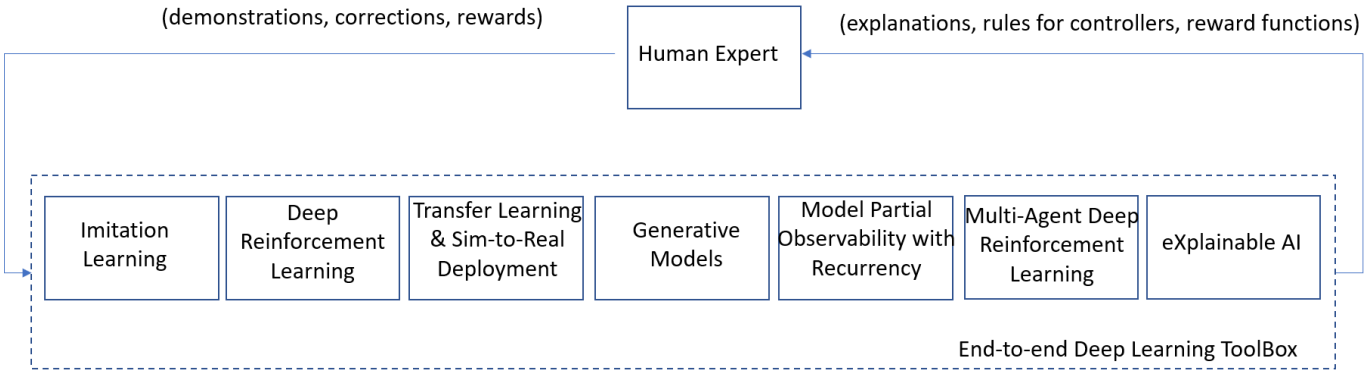


Fig. 1. End-to-end learning AI toolbox with Human-in-the-Loop for the autonomous management of programmable edge networks.

anomalous behavior or signs of performance degradation and initiate appropriate mitigation actions. These tasks typically require context-aware and temporally extended decision-making, making them particularly well-suited to Deep Reinforcement Learning (DRL) approaches [8], empowered by Generative AI (GAI) techniques [9].

III. End-to-End Learning AI Toolbox

To achieve autonomous edge network management in complex and dynamic scenarios, a data-driven AI toolbox is essential, combining human expertise with advanced AI to enable end-to-end learning and potentially surpass human performance. This section describes the toolbox components and their potential synergies: imitation learning from human administrators (Section III-A); DRL in simulated environments for safe, scalable policy refinement (Section III-B); transfer learning and sim-to-real deployment to align virtual training with real-world use (Section III-C); safe RL for robust decision-making (Section III-D); recurrent policies for partial observability in decentralized, resource-constrained settings (Section III-E); coordination among DRL agents for distributed intelligence (Section III-F); eXplainable AI (XAI) to ensure interpretability and trust (Section III-G); Generative AI (GAI) with LLMs to enhance perception, reasoning, and decision-making in complex networks (Section III-H); and synergies of toolbox elements toward autonomous end-to-end learning for next-generation wireless networks (Section III-I).

A. Imitation Learning from Human Administrators

The artificial agents should initially learn about initial management policies from the human (e.g., network administrator) [10]. This training process can be achieved through imitation learning, which can be cast as a supervised learning task.

Training data can be drawn from various sources, including network logs and events, historical SDN controller rule sets, and configuration scripts from Network Management Systems. These data sources provide hints in how experienced administrators respond to different network states and conditions.

To properly exploit imitation learning in the networks domain, several challenges must be addressed. One major issue is known as covariate shift, where the agent encounters states during deployment that differ from those seen during training, potentially leading to policy divergence. Additionally, the training data may suffer from sparse coverage, especially in the case of rare but critical situations, such as DDoS attacks or cascading link failures. There is also often a need for corrections, as early-stage agents may make poor decisions that require expert intervention to label or correct.

To mitigate the covariate shift, several techniques can be applied. DAgger (Dataset Aggregation) allows iterative refinement of the policy by incorporating expert corrections during agent policy rollouts. Online imitation learning frameworks enable continuous updates based on operator feedback, while active learning strategies prioritize data collection in uncertain or high-impact regions of the state space. In [11] is proposed an imitation learning framework between a transmitter and receiver devices using a radio channel for communication between them. In the framework, the transmitter acts as a teacher with access to the optimal control policies derived from the Digital Twin (DT), which accurately models the physical wireless channel environment. Additionally, the receiver acts as a learner, aiming to replicate or approximate these optimal control policies by learning from the semantic information transmitted from the transmitter device. This transmitter-receiver interaction effectively shifts wireless communication from mere data transfer to intelligent knowledge sharing, enabling semantic-aware, causally grounded, and resource-efficient distributed learned control in future DT-simulated networks. However, with this solution, the transfer of binary execution files can result in corruption as certain bits arrive with errors at the receiver. To mitigate this issue, complementary error-recovery techniques should be employed, such as Forward Error Correction (FEC) based on Hamming codes [12], Automatic Repeat Request (ARQ) protocols [13], or neural error correction methods [14].

B. Deep Reinforcement Learning in Simulation

Multi-armed Bandit (MAB) approaches to network management assume a stateless environment, where agents choose actions based solely on immediate rewards [15], [16]. While this simplifies decision-making, it prevents the agent from

accounting for how actions affect future network conditions, limiting long-term adaptability.

To improve decision quality, Contextual MAB extends this model by incorporating side information, such as, time of day, user profile, or device configuration, before the agent selecting an action [17]. These exogenous features help agents tailor decisions to current conditions. However, Contextual MAB still cannot capture how the environment evolves due to the agent's actions, making it insufficient for dynamic domains.

When the context includes features influenced by prior actions (e.g., traffic load, latency, or topology), RL offers a better fit [18]. RL models the problem as a Markov Decision Process (MDP), enabling agents to learn policies that optimize long-term outcomes by understanding state transitions. Still, traditional RL relies heavily on manual feature engineering, which is laborious and domain-specific.

Deep Reinforcement Learning (DRL) overcomes this challenge by leveraging deep neural networks to learn high-level representations directly from raw data, thereby eliminating the need for manually engineered features [8]. This end-to-end learning capability makes DRL particularly well-suited for dynamic and complex network environments, where state spaces are often high-dimensional and difficult to model explicitly. In such scenarios, DRL's ability to adapt to changing conditions and continuously improve its policy is invaluable for scalable and adaptive network management. Several state-of-the-art DRL algorithms are well-suited to this setting. For instance, policy gradient methods enable fine-grained control of decision-making by directly optimizing the policy. Proximal Policy Optimization (PPO) offers a balance between sample efficiency and stability, while Group Relative Policy Optimization (GRPO) extends this idea to collaborative or distributed agent scenarios, making it particularly promising for edge computing and distributed network intelligence. Collectively, these DRL algorithms can play a crucial role in autonomously optimizing innovative decision-making processes, significantly enhancing the performance and adaptability of future communication networks [19].

In a typical DRL setting, the agent operates over a formalized model comprising states, actions, and rewards. The state captures the current condition of the network and may include features, such as link utilization, flow statistics, and routing configurations. Based on this state, the agent selects an action, which may involve modifying flow rules, adjusting link weights, or changing controller policies. The agent then receives a reward that quantifies the desirability of the outcome, typically reflecting improvements in metrics like latency, throughput, resilience to failure, or policy compliance.

Simulation offers a safe, controllable setting for agents to explore diverse conditions without impacting live networks. It also enables repeatability and domain randomization to enhance real-world generalization.

1) Hierarchical Learning: Many SDN tasks lend themselves to a natural decomposition into sub-tasks, such as separating route selection from queue management or fault detection from recovery strategies. Hierarchical Reinforcement Learning (HRL) leverages this structure by organizing the policy into multiple layers, each responsible for different levels of abstraction. Higher-level policies can set strategic goals, such as selecting a traffic engineering strategy, while lower-level sub-policies handle operational details, such as selecting specific paths or modifying flow tables. This modular approach not only improves scalability and training efficiency but also facilitates policy reuse across different network scenarios.

2) Handling Rare Events with Domain Randomization: In real-world SDN environments, agents must manage both routine network dynamics and rare, high-impact events such as failures, anomalies, or attacks. To enhance robustness, domain randomization can be used during training by systematically varying simulated conditions, such as link failures, traffic surges, and topology changes. This diversity of experience helps agents generalize more effectively to unforeseen scenarios encountered during deployment.

C. Transfer Learning and Sim-to-Real Deployment

Policies trained in simulation environments often face significant challenges when deployed in real networks due to the well-known *sim-to-real gap*. Differences in topology fidelity, traffic patterns, timing behavior, and unmodeled hardware-specific phenomena can all degrade policy performance. To address this, several techniques can be employed to facilitate the transfer of learned policies to real-world deployments.

One approach is domain adaptation, which focuses on aligning the statistical properties of features observed in simulation with those encountered in the real environment. Techniques such as adversarial learning can be used to reduce the divergence between the two domains, thereby improving generalization. Another method is fine-tuning, in which the agent gradually adapts its policy using small amounts of real-world data collected under controlled, safe conditions. This incremental learning process helps mitigate risks while improving policy performance *in situ*. A third strategy is meta-learning, which aims to train agents that can rapidly adapt to new network environments or topologies with minimal additional training. By learning how to learn, such agents are better equipped to generalize across diverse operational conditions with limited feedback [11].

Collectively, these techniques provide a toolkit for bridging the gap between simulation and deployment [11], ensuring that learning-based SDN agents remain effective in realistic, dynamically evolving networks.

D. Safe Reinforcement Learning

To ensure the safe operation of SDN infrastructures, especially during training and early deployment phases, RL agents must operate under well-defined constraints. This is particularly important in production networks, where uncontrolled exploration may lead to service degradation, security breaches, or violation of service-level agreements.

One effective approach is the use of Constrained Policy Optimization [20], which explicitly restricts the agent's action

space to avoid decisions that could breach critical operational constraints. In the networks scenario, this could include guaranteeing minimum bandwidth for essential services or avoiding loops and black holes in routing. In high-risk or unfamiliar states, agents can fall back on backup policies, which consist of conservative, human-verified behaviors that serve as a safety net. Additionally, reward shaping can be employed to discourage unsafe behavior by introducing heavy penalties for actions that risk destabilizing the network or violating operational guarantees.

These safety mechanisms are not mutually exclusive and are often combined to form a layered safety architecture, enabling RL to be applied responsibly in real-world SDN deployments.

E. Modeling Partial Observability with Recurrent Policies

In practical SDN environments, agents often operate under partial observability. Due to factors like monitoring delays, state aggregation, or limited instrumentation, the full network state is not always available at decision time. This uncertainty can significantly impact the performance of learning-based agents if not properly addressed.

A suitable formalism for such scenarios is the Partially Observable Markov Decision Process (POMDP) [18], which extends traditional RL, incorporating hidden data. To operate effectively under this model, agents must maintain an internal representation or memory of past observations and actions.

Recurrent Neural Networks (RNNs), including architectures such as Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTM) networks, and Transformers [21], enhance an agent's capacity to maintain memory. These models enable agents to capture temporal dependencies and construct internal belief states, which are crucial for making informed decisions in environments with partial observability. Incorporating recurrence into the policy architecture equips agents to better manage the uncertainty and dynamics characteristic of real-world SDN operations.

F. Multi-Agent Deep Reinforcement Learning

In distributed SDN environments, spanning multiple administrative domains or using hierarchical control, multiple learning agents manage distinct network segments, while ensuring coherence on the global network operation. For instance, high-level intent management for persistent communication between two devices, apart each other from several network domains, requires each agent to make local, learned forwarding decisions toward a consistent end-to-end routing path between those devices.

To address the above challenges, Multi-Agent Deep Reinforcement Learning (MADRL) paradigm can be a valid solution [22], [23]. MADRL uses deep networks to handle high-dimensional, partially observable states. A key paradigm here is Centralized Training with Decentralized Execution (CTDE) [22], where agents learn cooperative policies using global state during training, but they act based on local observations at runtime. To enhance scalability and coordination, learning communication helps agents decide when, what, and whom to share information with, while thoughtful reward design ensures local actions support global network intents [22].

G. Explainable AI for Network Operators

For learning-based SDN agents to be trusted and adopted in operational settings, they must be able to provide clear and comprehensible justifications for their decisions. This is particularly important when actions impact critical network services, and operators need to validate the reasoning behind agents' decisions. The investigation on these aspects is aggregated in the area of Explainable AI (XAI) [10], [24]–[28].

In a recent work, the authors of [24] proposed a new taxonomy for organizing explainability methods in RL agents, based on the primary rationale that the agent conveys to the human network administrator when justifying its decisions. This taxonomy consists of three main classification groups: i) Feature Importance (FI); ii) Learning Process and Markov Decision Process (LPM); and iii) Policy-Level (PL). The FI methods give justifications rooted in the current input state of the system, identifying features that directly influenced the agent's most recent decision. The LPM methods offer explanations based on the agent's learned knowledge, typically acquired through prior training, either offline from historical network data or online via a digital twin model operating in parallel with the live network. Finally, the PL methods deliver explanations grounded in the agent's long-term policy behavior, summarizing how past experiences and strategies have shaped recent agent decisions. In [10] is explored the four phases for the deployment of Human-In-The-Loop (HITL) RL systems: initial agent development; agent learning; agent evaluation; and agent deployment. They also analyze where to apply explainability in HITL RL.

The XAI methods collectively support greater transparency, enabling operators to debug, validate, and ultimately trust the behavior of intelligent SDN agents.

H. Generative AI

RL enables agents to learn policies through trial and error. In contrast, Generative AI (GAI) models can produce context-specific policies by leveraging prior training, often without requiring additional online learning. GAI models aim to capture the full data distribution to produce new, realistic data samples. Although GAI models are computationally intensive, they offer advantages such as handling incomplete data and adapting management policies to real-time, partially observed network conditions. Considering these interesting capabilities, the GAI also belongs to the toolbox of Fig. 1, offering the next discussed advantages. First, GAI can give clear explanations about past autonomous management decisions made by agents, enabling XAI information to be sent and analyzed by an human expert. Second, GAI can adjust previous controller rules and create new rules with higher efficiency. Finally, GAI can tune the reward functions of DRL agents, inferring from learning past reward functions impact on the network performance.

Recently, RL-based Large Language Models (LLMs) like ChatGPT, DeepSeek, and Grok-3 have gained significant attention for their advanced multimodal data understanding [19]. A comprehensive review [29] has traced the evolution of LLMs from reasoning tasks to autonomous AI agents in domains like software engineering, math problem-solving, and multimedia. Integrating LLMs with wireless networks presents a promising approach to enhance overall network efficiency, as LLMs bring robust reasoning and decision-making capabilities [19]. This synergy can significantly improve the performance of future wireless communication systems by optimizing network resource management via network slicing [30], boosting transmission efficiency through semantic communication and Integrated Sensing and Communication (ISAC) [31], and achieving seamless, high-quality connectivity with Reconfigurable Intelligent Surfaces (RISs) [32].

I. Autonomous Learning in Next-Generation Mobile Networks

The proposed AI toolbox (Fig. 1) offers a unified framework to enhance future network management. Imitation learning catalyzes agent training from expert behavior [10], while DRL enables adaptive decision-making in complex and dynamic environments [8]. Transfer learning and sim-to-real techniques improve learning generalization [11].

To address partial observability [18], recurrent models provide memory over time [21], enhancing decision quality. MADRL orchestrates robust learning among agents [22]. Generative models excels training by simulating traffic or rare scenarios [9]. Finally, explainable AI (XAI) ensures transparency and trust in critical decisions [24]. Together, these techniques enable efficient, autonomous, and resilient operation of edge networking domains.

IV. Practical Limitations

Due to space constraints, this section focuses on the practical limitations of the most recent addition to the AI toolbox: Generative AI (GAI). The network edge is formed by switches, access points, or local domain controllers. Deploying intelligent agents based on online GAI [33] at the network edge domains offers significant benefits, including reduced response times, enhanced bandwidth management, and autonomous fault recovery in scenarios where centralized intelligent agents are temporarily unreachable. However, the edge devices are often resource-constrained, limiting their ability to host and execute large-scale neural models, which contain billions of parameters and require substantial computational resources. As a result, LLMs are typically deployed in remote datacenters, while lightweight neural models operate at the network periphery. This architectural separation among models requires for a collaborative framework in which edge AI models and remote LLMs work in tandem to support, under uncertainty, robust observation, inference, and decision-making. Recent studies underscore the importance of such hybrid intelligence architectures to address the challenges of real-time autonomous orchestration and resilience in edge networks [34]–[38].

In [38] is proposed a hybrid intelligence architecture that leverages an edge-hosted vector database to cache previous LLM-generated query responses. This enables low-latency replies for similar future requests, reducing redundant computation. When no cache hit occurs, requests are forwarded to cloud-based LLMs. To optimize this decision process, they model it as a Markov Decision Process and introduce DLRS, a diffusion-based multi-agent RL algorithm. DLRS extracts semantic features via a diffusion policy network to coordinate LLM request scheduling across edge nodes. Experiments show DLRS improves QoS, cutting latency and resource usage by *15%* compared to heuristics. Besides text, AI-generated frameworks at the network edge can also generate other novel content, namely, multimedia or 3D content [39].

Despite recent progress, several challenges remain in leveraging LLMs to enhance edge network performance for AI-generated content services [40]. In IoT-driven environments, data variability, from structured sensor streams to unstructured natural language, complicates semantic caching and vector similarity-based retrieval [38]. Determining when a cached LLM-generated response is semantically appropriate for a new request remains an open issue. Long-term caching at the edge also introduces complexity, requiring intelligent data eviction and retention policies to balance freshness, relevance, and diversity. Furthermore, many LLM caching and scheduling strategies have been evaluated in idealized conditions, raising concerns about their robustness in heterogeneous edge environments with diverse hardware, dynamic network states, and variable sensor activity. These challenges underscore the need for adaptive, context-aware orchestration techniques for scalable, real-time delivery of AI-generated content [40].

Overcoming the above challenges is crucial for enabling the practical deployment of GAI in programmable edge networks. Future work must investigate methods for semantic abstraction and generalization across diverse request types, possibly integrating more sophisticated language understanding models into the feature extraction pipeline. Additionally, dynamic cache management policies should incorporate aging, utility, and diversity metrics to maintain a responsive and compact edge vector database, caching autonomously learned network management decisions. Lastly, training RL agents via transfer learning can apply robust policies across network domains. In addition, GAI can revolutionize the knowledge on self-managing edge networks [41], including future mobile networks [42] for optimizing their operation [9], communications security [43], or AI-generated Service Content [39].

V. Conclusion and Future Work

This paper presents an end-to-end learning roadmap for Artificial Intelligence (AI)-powered autonomous edge network management, integrating Imitation Learning, Reinforcement Learning, Generative AI, and eXplainable AI. The proposed AI toolbox can enhance network efficiency through adaptive, real-time, and context-aware decision-making. In dynamic, resource-constrained edge environments, lightweight AI agents

can enable predictive and energy-efficient management, improving responsiveness and service quality. The agents self-adapt to evolving user demands and network conditions, supporting resilient and scalable operations. Future research should address challenges such as multi-agent coordination, sim-to-real policy transfer, and secure, accountable learning to fully realize intelligent and autonomous edge networking.

VI. Acknowledgment

José Moura was funded by FCT/MECI through national funds and when applicable co-funded EU funds under UID/50008: Instituto de Telecomunicações. The work of Pedro Santana was partially supported by FCT under ISTAR projects UIDB/04466/2020 and UIDP/04466/2020.

References

[1] V. Chamola *et al.*, "Future of connectivity: A comprehensive review of innovations and challenges in 7g smart networks," *IEEE Open Journal of the Communications Society*, 2025.

[2] M. S. Akbar *et al.*, "On challenges of sixth-generation (6g) wireless networks: A comprehensive survey of requirements, applications, and security issues," *Journal of Network and Computer Applications*, p. 104040, 2024.

[3] C.-X. Wang *et al.*, "On the road to 6g: Visions, requirements, key technologies, and testbeds," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 905–974, 2023.

[4] D. C. Nguyen *et al.*, "6g internet of things: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359–383, 2021.

[5] W. Jiang *et al.*, "The road towards 6g: A comprehensive survey," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 334–366, 2021.

[6] D. Kreutz *et al.*, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[7] M. Talal *et al.*, "A comprehensive systematic review on machine learning application in the 5g-ran architecture: Issues, challenges, and future directions," *Journal of Network and Computer Applications*, p. 104041, 2024.

[8] S. A. Ullah *et al.*, "From nodes to roads: Surveying drl applications in mec-enhanced terrestrial wireless networks," *IEEE Communications Surveys & Tutorials*, 2025.

[9] F. Khoramnejad and E. Hossain, "Generative ai for the optimization of next-generation wireless networks: Basics, state-of-the-art, and open challenges," *IEEE Communications Surveys & Tutorials*, 2025.

[10] C. O. Retzlaff *et al.*, "Human-in-the-loop reinforcement learning: A survey and position on requirements, challenges, and opportunities," *Journal of Artificial Intelligence Research*, vol. 79, pp. 359–415, 2024.

[11] C. K. Thomas, W. Saad, and Y. Xiao, "Causal semantic communication for digital twins: A generalizable imitation learning approach," *IEEE Journal on Selected Areas in Information Theory*, vol. 4, pp. 698–717, 2023.

[12] S. Lin and D. J. Costello Jr., *Error control coding: Fundamentals and Applications*. Englewood Cliffs, New Jersey, USA: Prentice-Hall, 1983, 0-13-283796-X.

[13] S. Lin, D. J. Costello, and M. J. Miller, "Automatic-repeat-request error-control schemes," *IEEE Communications magazine*, vol. 22, no. 12, pp. 5–17, 1984.

[14] A. Esposito, S. Rampone, and R. Tagliaferri, "A neural network for error correcting decoding of binary linear codes," *Neural networks*, vol. 7, no. 1, pp. 195–202, 1994.

[15] P. Santana and J. Moura, "A bayesian multi-armed bandit algorithm for dynamic end-to-end routing in sdn-based networks with piecewise-stationary rewards," *Algorithms*, vol. 16, no. 5, 2023. [Online]. Available: https://www.mdpi.com/1999-4893/16/5/233

[16] A. Slivkins *et al.*, "Introduction to multi-armed bandits," *Foundations and Trends® in Machine Learning*, vol. 12, no. 1-2, pp. 1–286, 2019.

[17] J. Langford and T. Zhang, "The epoch-greedy algorithm for multi-armed bandits with side information," *Advances in neural information processing systems*, vol. 20, 2007.

[18] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[19] Y. Qiao *et al.*, "Deepseek-inspired exploration of rl-based llms and synergy with wireless networks: A survey," *arXiv preprint arXiv:2503.09956*, 2025.

[20] J. Achiam *et al.*, "Constrained policy optimization," in *International conference on machine learning*. PMLR, 2017, pp. 22–31.

[21] B. Lin *et al.*, "Tite: A transformer-based deep reinforcement learning approach for traffic engineering in hybrid sdn with dynamic traffic," *Future Generation Computer Systems*, vol. 161, pp. 95–105, 2024.

[22] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: a survey," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 895–943, 2022.

[23] T. Rashid *et al.*, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 178, pp. 1–51, 2020.

[24] S. Milani *et al.*, "Explainable reinforcement learning: A survey and comparative review," *ACM Computing Surveys*, vol. 56, no. 7, pp. 1–36, 2024.

[25] E. Puiutta and E. M. Veith, "Explainable reinforcement learning: A survey," in *International cross-domain conference for machine learning and knowledge extraction*. Springer, 2020, pp. 77–95.

[26] S. Atakishiyev *et al.*, "Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions," *IEEE Access*, 2024.

[27] C. Rudin *et al.*, "Interpretable machine learning: Fundamental principles and 10 grand challenges," *Statistic Surveys*, vol. 16, pp. 1–85, 2022.

[28] A. Verma *et al.*, "Programmatically interpretable reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 5045–5054.

[29] M. A. Ferrag, N. Tihanyi, and M. Debbah, "From llm reasoning to autonomous ai agents: A comprehensive review," *arXiv preprint arXiv:2504.19678*, 2025.

[30] L. Lei *et al.*, "Dynamic-adaptive ai solutions for network slicing management in satellite-integrated b5g systems," *IEEE Network*, vol. 35, no. 6, pp. 91–97, 2022.

[31] G. Zhu *et al.*, "Semantic-based channel state information feedback for uav-assisted isac systems," *IEEE Internet of Things Journal*, 2024.

[32] Y. Liu *et al.*, "Reconfigurable intelligent surfaces: Principles and opportunities," *IEEE communications surveys & tutorials*, vol. 23, no. 3, pp. 1546–1577, 2021.

[33] M. Sheraz *et al.*, "A comprehensive survey on genai-enabled 6g: Technologies, challenges, and future research avenues," *IEEE Open Journal of the Communications Society*, 2025.

[34] Z. Hao *et al.*, "Hybrid slm and llm for edge-cloud collaborative inference," in *Proceedings of the Workshop on Edge and Mobile Foundation Models*, 2024, pp. 36–41.

[35] Y. Tian *et al.*, "An edge-cloud collaboration framework for generative ai service provision with synergetic big cloud model and small edge models," *IEEE Network*, 2024.

[36] Y. Yang *et al.*, "Human-centric autonomous systems with llms for user command reasoning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 988–994.

[37] M. Zhang *et al.*, "Edgeshard: Efficient llm inference via collaborative edge computing," *IEEE Internet of Things Journal*, 2024.

[38] Z. Yao *et al.*, "Enhancing llm qos through cloud-edge collaboration: A diffusion-based multi-agent reinforcement learning approach," *IEEE Transactions on Services Computing*, pp. 1–17, 2025.

[39] M. Xu *et al.*, "Unleashing the power of edge-cloud generative ai in mobile networks: A survey of aigc services," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 2, pp. 1127–1170, 2024.

[40] H. Du *et al.*, "Diffusion-based reinforcement learning for edge-enabled ai-generated content services," *IEEE Transactions on Mobile Computing*, vol. 23, no. 9, pp. 8902–8918, 2024.

[41] S. Sai, M. Kanadia, and V. Chamola, "Empowering iot with generative ai: applications, case studies, and limitations," *IEEE Internet of Things Magazine*, vol. 7, no. 3, pp. 38–43, 2024.

[42] A. Karapantelakis *et al.*, "Generative ai in mobile networks: A survey," *Annals of Telecommunications*, vol. 79, no. 1, pp. 15–33, 2024.

[43] C. Zhao *et al.*, "Generative ai for secure physical layer communications: A survey," *IEEE Transactions on Cognitive Communications and Networking*, 2024.