Array 14 (2022) 100136

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array

Knowledge-defined networking: Applications, challenges and future work

Sepehr Ashtari [a,*], Ian Zhou [a], Mehran Abolhasan [a], Negin Shariati [a], Justin Lipman [a], Wei Ni [b]

[a] University of Technology, Sydney, Australia
[b] The Commonwealth Scientific and Industrial Research Organisation, Sydney, Australia

ARTICLE INFO

Keywords:
6G
Knowledge-defined networking (KDN)
Software-defined networking (SDN)
Open-radio access network (O-RAN)
Localization
Machine learning
Mobility coordination
Network management
Resource management

ABSTRACT

Future 6G wireless communication systems are expected to feature intelligence and automation. Knowledge-defined networking (KDN) is an evolutionary step toward autonomous and self-driving networks. The building blocks of the KDN paradigm in achieving self-driving networks are software-defined networking (SDN), packet-level network telemetry, and machine learning (ML). The KDN paradigm intends to integrate intelligence to manage and control networks automatically. In this study, we first introduce the disadvantages of current network technologies. Then, the KDN and associated technologies are explored with three possible KDN architectures for heterogeneous wireless networks. Furthermore, a thorough investigation of recent survey studies on different wireless network applications was conducted. The aim is to identify and review suitable ML-based studies for KDN-based wireless cellular networks. These applications are categorized as resource management, network management, mobility management, and localization. Resource management applications can be further classified as spectrum allocation, power management, quality-of-service (QoS), base station (BS) switching, cache, and backhaul management. Within network management configurations, routing strategies, clustering, user/BS association, traffic classification, and data aggregation were investigated. Applications in mobility management include user mobility prediction and handover management. To improve the accuracy of positioning in indoor environments, localization techniques were discussed. We classify existing research into the respective KDN architecture and identify how the knowledge obtained will enhance future networks; as a result, researchers can extend their work to empower intelligence and self-organization in the network using the KDN paradigm. Finally, the requirements, motivations, applications, challenges, and open issues are presented.

1. Introduction

Fifth-generation (5G) wireless communication systems provide higher data rates, massive connectivity, and low-latency communication. However, the current 5G cellular architecture lacks intelligence and sufficient flexibility to handle massive machine-type communication (mMTC), low-latency, and enhanced mobile broadband (eMBB) [1]. The sixth-generation (6G) cellular network is a promising technology to address 5G shortcomings. To achieve this, 6G will enable greater intelligence within the network to overcome a number of challenges and improve performance [2]. As a result, an architectural transformation is required for 5G to 6G cellular networks.

The concept of the knowledge plane (KP) was introduced by Clark et al. [3]. As shown in Fig. 1 KP is an additional plane over a network with inbuilt machine learning (ML) capabilities. The incorporation of KP in software-defined networking (SDN) architecture is referred to as knowledge-defined networking (KDN) [4], where knowledge is the processed network information using an ML algorithm. Knowledge is used for recommendation and automation across different applications in wired and wireless networks. Therefore, knowledge can be referred to as intelligence over a network, and having intelligence over a network with different environmental characteristics can be a breakthrough in network performance. For instance, in resource management problems, parameters such as bandwidth, quality of service (QoS), and power can be obtained and processed using an ML algorithm in different network situations. The output of ML is then stored as knowledge for network automation. Moreover, in networking applications, routing decisions can benefit from knowledge for better route discovery while the network is overpopulated. Further, user information, including mobility patterns and velocity, can be used as an initial stage to generate knowledge to improve the accuracy of localization and handover.

KDN can also be referred to as an application of autonomous networking. The concept of an autonomous network comes from the growth of ML and artificial intelligence (AI); for instance, in self-driving cars, an ML agent will run the car without a human operator. Similarly,

* Corresponding author.
E-mail address: sepehr.ashtarinakhaei@student.uts.edu.au (S. Ashtari).

https://doi.org/10.1016/j.array.2022.100136
Received 29 September 2021; Accepted 22 February 2022
Available online 4 April 2022

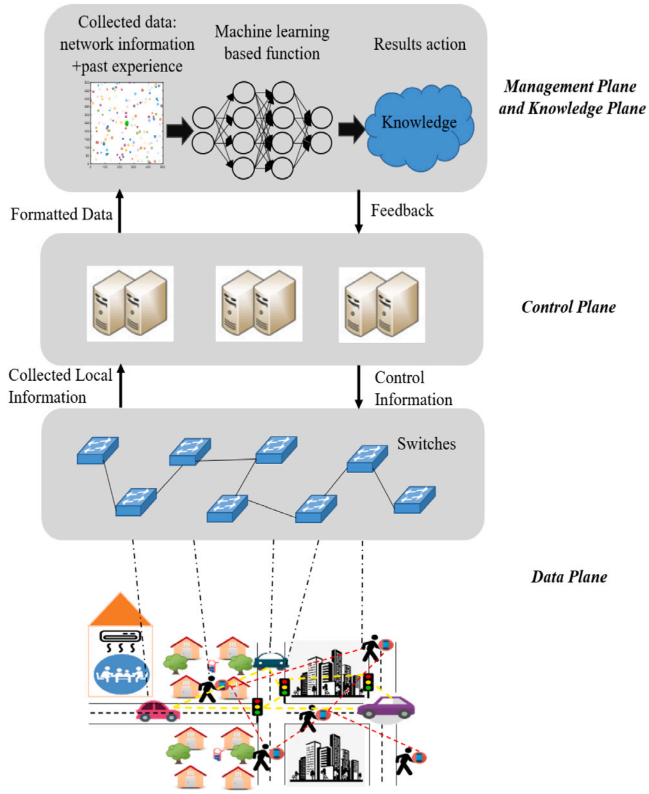


Fig. 1. Knowledge-defined orchestration in wireless networks.

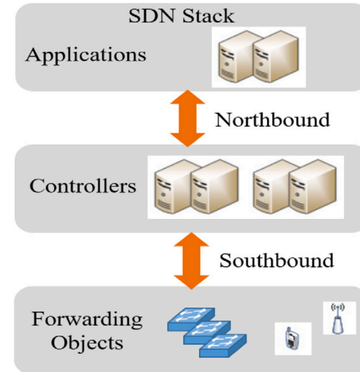


Fig. 2. The SDN northbound and southbound interfaces.

an autonomous network can manage and optimize network applications without human intervention. In autonomous networks, network information or telemetry is collected and used by ML techniques to automatically troubleshoot, instruct, or manage the network. Hence, monitoring and retrieval of network telemetry data in real time will provide an opportunity for ML-based optimization algorithms to enable intelligence for 6G networks.

The fundamental building blocks in KDN are network telemetry, SDN, and ML. Network telemetry is network information, such as NetFlow data, sFlow data, queue occupancy, policy rules, and processing time. However, to fully auto-mate the network, information such as hop latency, link utilization, packet drop, and queue congestion states are also required. This temporary data is available through in-band network telemetry (INT) or packet-level network states. The packet-level network state information can be collected using a new southbound domain-specific language called P4 [5], which allows the collection of information directly from the data plane. As shown in Fig. 2 the southbound and northbound interfaces are part of the SDN communication protocol, where the southbound interface allows communication between controllers and switches, and other low-level network components. The northbound interface enables communication with high-level components [6]. The next building block of KDN is SDN, which enables the global network view, network programmability functions, and flexibility to manage the network. The combination of network analytics and SDN provides a foundation for the KDN paradigm. However, an ML algorithm will be the heart of KDN, meaning that an ML technique can provide an efficient and optimized strategy to operate the network autonomously.ML is a key component of the KDN paradigm that provides a solution given the telemetry data.

One of the indications of the KDN concept is the open radio access network (O-RAN). O-RAN is an emerging technology that enables service heterogeneity, on-demand service deployment, and simultaneous coordination of heterogeneous devices [7]. Over the past few years, various efforts from the research industry have been made to enhance the radio access network (RAN) [7,8]. Among them, cloud-RAN or centralized-RAN (C-RAN) is a promising RAN architecture that helps to reduce base-band expenses. Further, a virtualized radio access network (V-RAN) brings the benefit of cloud and virtualization to increase the network's agility, scalability, and flexibility. This architecture has helped RAN with new opportunities for virtualization and cost reduction. V-RAN simplified the management of RAN devices and deployment. Although C-RAN and V-RAN are both cost-effective and readily available to any changes to service requirements, they lack openness to maximize the benefit of virtualization. To overcome the limitations associated with both C-RAN and RAN, O-RAN accommodates the baseband unit (BBU) and remote radio unit (RRU) software and hardware from different vendors [9]. O-RAN is open hardware with an operator-defined RAN architecture that provides intelligent radio control for future cellular networks. The other innovative technology that is going to help accommodate intelligence in the network is disaggregating the traditional control and data plane of the RAN to introduce an RAN intelligent controller (RIC) [10]. O-RAN uses the SDN and network function virtualization (NFV) principles and apply the ML algorithm to create automation in cellular networks. O-RAN standardizes the control plane using open infrastructure and provides programmability. These are clear indications of demand for KDN architecture to simplify the process by adding the KP and incorporating ML algorithms to empower network intelligence with ML-enabled applications [11].

Owing to the evolution of ML applications in wireless networks and the presence of SDN, less effort has been made to survey related studies that are compatible with KDN architecture. The authors of [4] introduced the basics and applications of ML algorithms in KDN to control and operate networks. They used SL to obtain an intermediate knowledge of the traffic load to perform network configurations. In [12], a deep RL with a convolutional neural network (CNN) in the context of KDN was utilized to make routing decisions based on QoS in complex networks. In [13], deep learning (DL) algorithms were proposed to produce three artificial intelligence (AI) modules to manage the scalability and energy consumption in data center network (DCN) systems. In this study, a hybrid optical/electronic DC network is designed with the management of the knowledge-defined network orchestrator (KD-NO) software module. The software module uses the KDN architecture and deploys three DL-based AI modules in the knowledge plane of the KD-NO to make intelligent decision-making and precise predictions of traffic volume, data latency, and hardware demands. In [14], the authors presented a new network architecture and networking solution for high user density in 5G. Their research utilizes SDN, KDN, and NFV, where they take advantage of ML in the context of KDN. Additionally, owing to the massive connectivity of devices, the concept of self-driving networks by applying KDN and network telemetry was proposed [15]. Although there are studies on

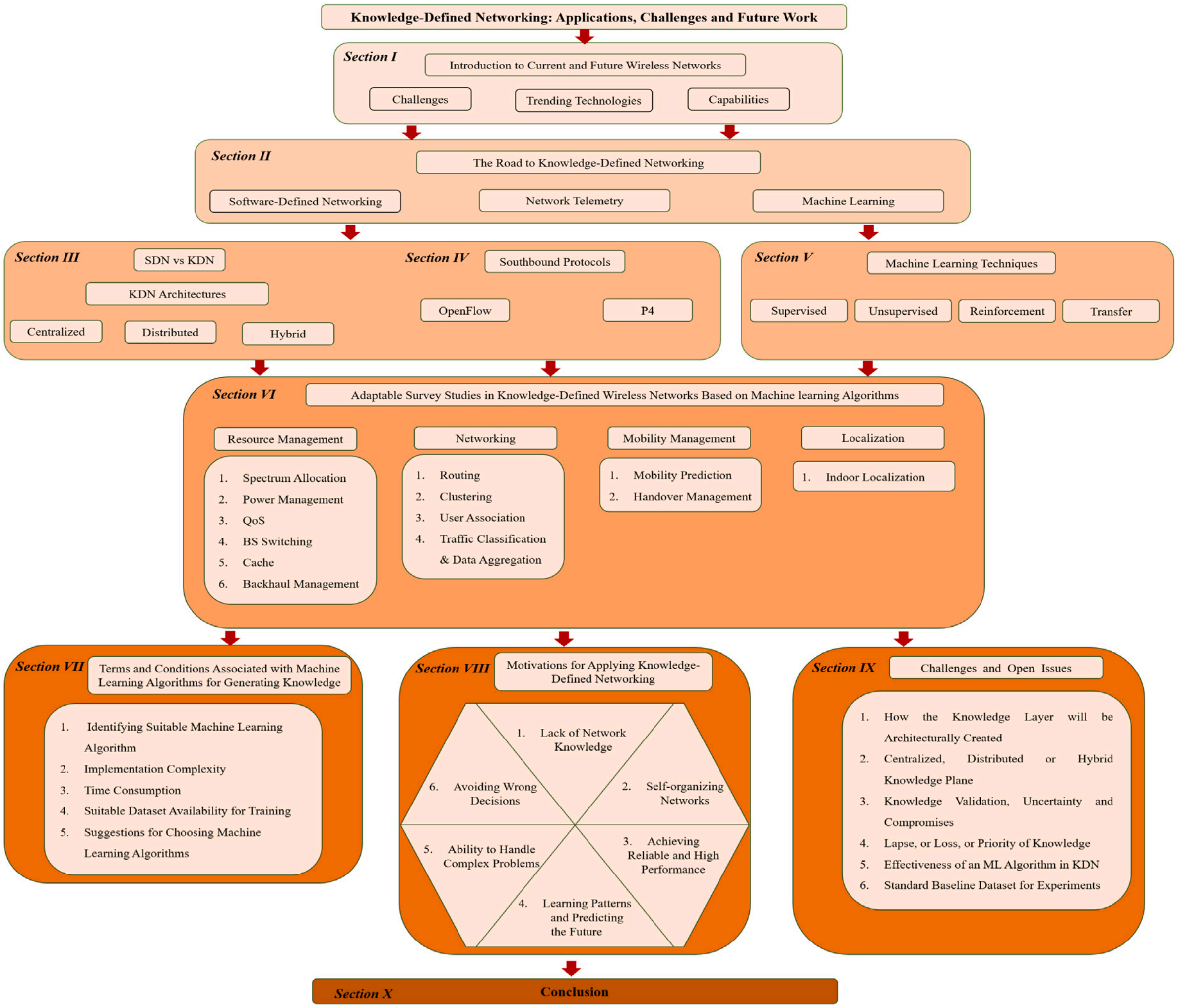


Fig. 3. The paper structure.

KDN, there is no survey that comprehensively studies applications of ML in different layers of wireless networking with their relationship and usage in the KDN architecture. Moreover, there exist some limitations in recent works and the overall challenges in applying KDN, which also need attention for KDN to be fully functional. Furthermore, the benefits of having a KDN in the network are clearly stated in the proposed research studies. For instance, the authors of [4,15] investigated the general advantages and use cases of the KP and introduced the KDN switch operator along with its architecture. More concretely, the work in [12] examined deep reinforcement learning (DRL) for QoS-aware routing. The literature in [13] proposed a solution for mitigating data center congestion issues by deploying KDN in the network. Such works demonstrate the benefits of adapting knowledge before making decisions in wireless networks.

The motivation of this study comes from the lack of intelligence and automated networks. In the near future, network congestion and smart applications will force the network to have self-managing systems. Considering the shortcomings of 5G and the continuous research activities in wireless networks to enable intelligence and automation in wireless networks, specifically in 6G, a more structured and comprehensive survey is required to understand the potential advantages of KDN-based ML networks. Specifically, we first introduce three possible KDN architectures adapted in wireless networks, including centralized, distributed, and hybrid. Then, we concentrate on the latest achievements in applications of ML from the MAC layer up to the application layer, including resource management, networking, mobility management, and localization. ML is the heart of the KDN and any future network that needs automation and self-organization must adapt the KDN-based ML algorithm. This is why we are conducting a comprehensive study about all the research conducted in the ML categories that suit network automation. The focus of the study is to realize the new ML achievements and take advantage of each to adapt the most compatible ML-based algorithm in networking problems. We then mapped the ML method to the most appropriate KDN architecture. The mapping aims to help researchers identify the target area and the relevant ML with the support of a suitable KDN architecture. This survey aims to offer a comprehensive analysis of KDN applications using ML-based studies to enable intelligence and automation for current and future wireless networks. Our contributions are summarized as follows:

(1) Providing a concise explanation about the shortcoming of wireless networks and potential solutions for network automation.
(2) An introduction to KDN and representation of the main building blocks in KDN, such as SDN, network telemetry, and ML algorithms, is delivered.
(3) A complete comparison discussion between SDN and KDN is offered.
(4) We introduce and analyze three possible architectures of KDN in wireless cellular networks.
(5) A comprehensive study of packet-level programming and southbound interfaces of SDN, including Open Flow (OF) and the domain-specific language (DSL), known as P4, is discussed.

Where, packet-level programming is used to perform ML algorithm on collected data and apply the ML-based instructions after the learning procedure.

(6) An overview of supervised learning (SL), unsupervised learning (UL), reinforcement learning (RL), transfer learning (TL), and neural networks (NNs) with basic principles and common applications in wireless networks is provided.

(7) A thorough review of the applications of ML within the KDN paradigm is presented, covering resource management, networking, mobility management, and localization. This survey focuses on the MAC layer, network layer, and application layer (the PHY layer is outside the scope of this study). To cover every aspect of a wireless network, resource management is broken down into resource allocation, power management, QoS, base station (BS) switching, cache, and backhaul management. Networking applications are further classified into route selection, clustering, user association, traffic classification, and data aggregation. Mobility prediction and handover management lie within the mobility section, and finally, indoor localization.

(8) Finally, the terms and conditions associated with ML algorithms for generating knowledge and motivations for applying KDN were identified. Then, a summary of the challenges and current issues in KDN is discussed, followed by the conclusion.

The remainder of this paper is organized as follows: In Section 2 KDN, the enabling technologies for KDN are explained, such as SDN, network telemetry, and ML. Then, in Section 3, SDN and KDN are compared, and the potential architecture of the KDN is illustrated. In Section 4, the southbound interfaces for packet-level programming are thoroughly studied, OF and P4, in addition to some of the disadvantages of current southbound interfaces and applications of P4. Section 5 describes some of the popular ML techniques utilized in KDN with a brief introduction of their applications in wireless communication networks. In Section 6, a complete study of resource management, networking, mobility management, and localization is presented. Later, in Section 7, terms and conditions associated with ML algorithms for generating knowledge are discussed, including identifying suitable ML algorithms, implementation complexity, time consumption, suitable dataset availability for training, and suggestions for choosing ML algorithms. Moreover, in Section 8, motivations for applying KDN are presented, including a lack of network knowledge, self-organizing networks, achieving reliable and high performance, learning patterns and predicting the future, ability to handle complex problems, and avoiding a wrong decision, followed by challenges and open issues in KDN in Section 9. Finally, Section 10 concludes the study. Fig. 3 depicts the survey's structure as a graphical representation for a better understanding of how the survey is shaped and analyzed according to each section. Further, for readers' convenience, Table 1 lists the abbreviations used throughout this survey.

2. The road to knowledge-defined networking

It is estimated that by 2030, the number of connected devices in cellular networks will reach 100 billion [16]. Furthermore, the majority of network occupancy is due to the demand for high-definition video data, which leads to massive data traffic [17–19]. Hence, many existing algorithms cannot process traffic flows, resulting in a loss of information. Moreover, algorithms are incapable of offering optimum system performance when the network environment is dynamic and random. Therefore, these algorithms are unable to meet the requirements of the 6G cellular networks. To overcome these problems and achieve better performance, researchers have developed optimization methods to attain effective solutions to get closer to optimal and suboptimal performance. However, many studies presume a static network environment rather than considering the random nature of networks [20,21]. Additionally, traditional centralized algorithms for network management and simultaneous collection of global data are affected as the number of connected devices increases [2]. Consequently, one viable solution is to use KDN to automatically optimize, diagnose, and troubleshoot the network [4]. To enable KDN, a centralized controller with ML capabilities must collect information from the network to create knowledge via an appropriate ML algorithm. The centralized controller for collecting telemetry data must have SDN functionalities.

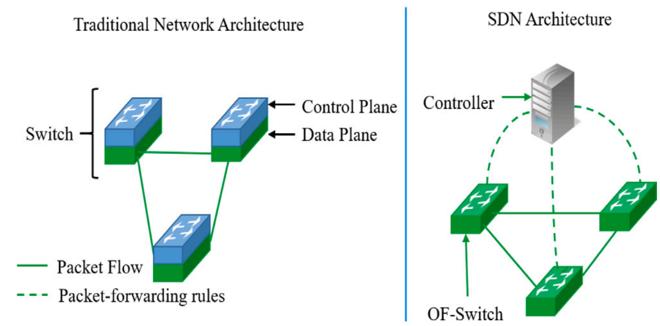


Fig. 4. Traditional network architecture versus SDN architecture.

Software-defined networking (SDN) is a new networking paradigm for decoupling data and control planes [22]. Decoupling these two planes enables SDN to operate as a centralized controller to manage the network. The global view of the SDN controller provides advantages such as network flexibility, programmability, and efficient management over the traditional network. In the conventional network architecture, network administrators are required to manually configure and troubleshoot switches and routers within their organization [23]. From Fig. 4, the difference between traditional network architecture and SDN can be seen. SDN provides network programmability within the control plane, whereas traditional networks do not offer any flexibility. Traditional networks rely only on the physical infrastructure to create connections and run the network. On the other hand, SDN enables users to use software programs instead of physical infrastructure to provision new devices [24].

In addition to network softwarization, adequate data collection is required in KDN to make appropriate and accurate decisions. With the new advancements in data plane elements, routers and switches are capable of computation and storage, which makes the network monitoring and network telemetry accessible [25]. Network telemetry provides flow information, real-time packet information, and other critical packet-level data, as well as network state monitoring and organization with centralized network analytics. Hence, network telemetry and network analytics present a richer view of network performance metrics, providing an extra advantage over conventional network management techniques. The incorporation of SDN and network analytics provides essential elements required by the KP. However, the last piece of the puzzle to make the KDN fully functional is to integrate ML. ML uses network telemetry and historical data to process and find valuable information about the network, where this information is stored as knowledge to improve network performance.

ML algorithms are generally classified as supervised, unsupervised, and reinforcement learning [26]. In SL, the learning agent learns using a dataset as an input vector and maps the inputs to the outputs based on the previous inputs with their provided outputs. The dataset is a collection of labeled samples, where each element is called a feature vector. In UL, the dataset is the collection of unlabeled samples, where the learning agent tries to categorize the input. Finally, in RL, the machine continuously observes the environment to improve the decision-making process. This technique continuously provides updated policies based on environmental feedback. Each ML algorithm can assist in the different applications of wireless networks. For instance, in SL, the KP learns the behavior of the network variables, such as network configuration and traffic load, which will enable the system to increase

Table 1
Abbreviations.

5G	5th generation	MBS	macro-base station
6G	6th generation	MDP	markov decision process
AI	artificial intelligence	ML	machine learning
ANN	artificial neural network	mMTC	massive machine type communication
AoA	angle of arrival	mmWave	millimeter wave
API	application programming interface	MPLS	multiprotocol label switching
ASIC	application specific integrated circuit	MRE	modified Roth–Erev
BBU	baseband unit	MS	mobile station
BM	behavioral model	NE	Nash equilibrium
BS	base station	NFV	network function virtualization
CHO	conditional handover	NLOS	non-line-of-sight
CNN	convolutional neural network	NN	neural network
CP	control Plane	OF	open flow
C-RAN	cloud radio access network	ONF	open network foundation
CRE	cell range extension	O-RAN	open-radio access network
CRN	cognitive radio network	OSPF	open shortest path first
CSI	channel state information	OVS	open vSwitch
D2D	device-to-device	PBS	pico-base station
DCN	data center network	PCA	principal component analysis
DL	deep learning	PISA	protocol independent switch architecture
DNM	deep neural model	POF	protocol oblivious forwarding
DNN	deep neural network	PU	primary user
DPDK	data plane development kit	PVS	POF switch
DPPO	distributed proximal policy optimization	QoE	quality of experience
DQN	deep Q-network	QoS	quality of service
DRL	deep reinforcement learning	RAN	radio access network
DSL	domain-specific language	RB	resource block
DT	decision tree	RIC	RAN intelligent controller
ECR	energy consumption ratio	RL	reinforcement learning
ELM	extreme learning machine	RNN	recurrent neural network
eMBB	enhance mobile broadband	RRH	remote radio head
eNB	evolved node B	RRU	remote Radio unit
ESN	echo state network	RSRP	reference signal received power
ETL	extract-transform-load	RSRQ	reference signal received quality
FBS	femto-base station	RSS	received signal strength
FC	femtocell	RSSI	received signal strength indicator
FFNN	feed-forward neural network	RSU	roadside unit
GF	gradient follower	SBS	small base station
gNB	gigabit node B	SCN	small cell network
GPU	graphics processing unit	SDN	software-defined networking
HAL	hardware abstraction librar	SINR	signal-to-interference-plus-noise
HetNet	heterogeneous network	SL	supervised learning
ICIC	inter-cell interference coordination	SMU	spectrum selection utility
IDE	integrated development environment	SNR	signal-to-noise ratio
INT	in-band network telemetry	SOM	self-organizing map
IoT	Internet of things	SON	self-organizing networks
IP	Internet protocol	SSU	spectrum selection utility
ISP	Internet service provider	SVD	singular value decomposition
IVN	intelligent vehicular networks	SVM	support Vector Machine
JSON	javaScript Object Notation	SU	secondary user
KDN	knowledge-defined networking	TL	transfer learning
KD-NO	knowledge defined-network orchestrator	TLV	type-length-value
KNN	k-nearest neighbors	TTP	table type patterns
KP	knowledge plane	UAV	unmanned aerial vehicle
LFU	least frequently used	UDN	ultra dense networks
LOS	line-of-sight	UE	user equipment
LRU	least recently used	UL	unsupervised learning
LTE	long-term evolution	V2I	vehicle to Infrastructure
LTE-U	LTE-unlicensed	V2V	vehicle-to-vehicle
M2M	machine-to-machine	VANET	vehicular ad hoc network
MAB	multi-armed bandit	V-RAN	virtual radio access network
MAC	medium access control	WMMSE	weighted minimum mean square error
MBM	modified Bushand Mostelle	WSN	wireless sensor network

the network performance once the features are fed to the algorithm. UL assists the network operator by following the correlations in the data. For example, ML may predict the mobility effect on a user's network link. Moreover, in RL, the learning algorithm will discover the best action that leads to an optimal configuration in a network. As RL adapts to the environment, it eventually determines the target policy. This approach returns the optimal action based on the target strategy and then applies them to the KDN controller to enable the best configuration. RL provides extensive benefits for resource management [27]. As a result of network softwarization, network telemetry, and integration of ML in the KP, the foundation of KDN has emerged, which will provide the advantage of overcoming the drawbacks of conventional resource management, mobility management, networking, and localization [4].

3. Introduction to knowledge-defined networking

Traditional cellular networks suffer from complexity, lack of flexibility, proprietary and expensive equipment, and compound control-plane protocols. However, applying the idea of running network applications to a logically centralized controller changed everything. SDN

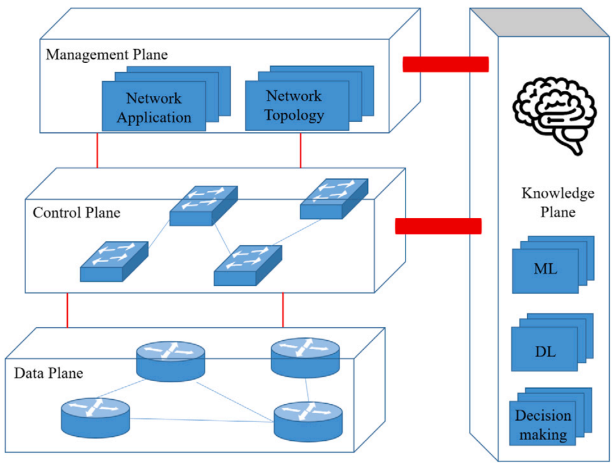


Fig. 5. KDN architecture.

architecture enables programmability, more straightforward configurations, and network management. To go one step forward in software-defined cellular networks, KDN introduced intelligence in addition to programmability and centralized control of SDN.

3.1. Software-defined networking versus knowledge-defined networking

The integration of the KP in SDN is a new concept called knowledge-defined networking (KDN). The concept of KP is to add one more plane to the traditional two planes of SDN. This new paradigm incorporates SDN, data analytics, and ML. The KDN paradigm has several advantages: first, it has a global view of the network, and second, it enables telemetry data to be collected by the management plane to transform the data into knowledge via ML. The knowledge will later turn into decisions by nodes to achieve efficient network operations [18,19]. The benefit of having the KDN over traditional networks is that it automatically operates based on the knowledge obtained from the network. Fig. 5 illustrates the KDN architecture, including the data plane, control plane, management plane, and KP.

The data plane in KDN is responsible for forwarding, dropping, processing, and packet modification. This layer works precisely like the data plane in SDN, where it consists of physical and virtual device elements. This layer operates unaware of the rest of the network and relies on the instructions and control rules coming from other planes.

The control plane exchanges information and updates the data plane processing and matching strategy rules. The logically centralized controller exchanges data and updates policies using a southbound application programming interface (API). The controller gathers the data and network state from the data plane and updates the flow tables to perform actions. In KDN, the data are also utilized to allow the KP to know which appropriate action is required. Then, the controller receives an action from the KP and updates the flow tables accordingly. These actions are usually used for forwarding and routing packets, while the data plane is populated.

The management plane facilitates network topologies, support services, and configuration of the network devices. This layer must ensure that the network operates fully with the maximum performance. This functionality of the network in KDN is handled by the centralized controller as well, being responsible for monitoring the data plane and observing network analytics. The network analytics will then be collected and stored as a network state and telemetry. This information is also monitored by the KP for possible updates of the network topology.

KP is the brain of the architecture and responsible for modeling network behavior and decision-making. These decisions are made for different applications in the network, such as resource management, networking configurations, mobility management, and localization. In this layer, knowledge is created by ML algorithms, and new policies are obtained.

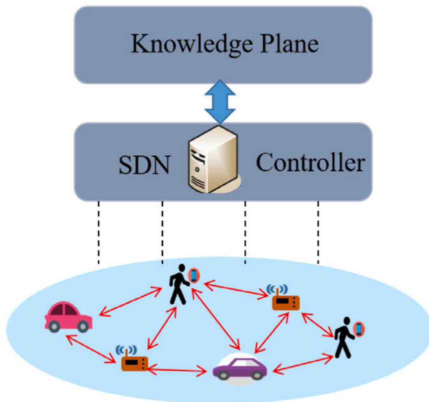
3.2. KDN architecture

A number of studies have introduced the KDN architecture. For instance, the authors of [28] restated the concept of the KP in the context of SDN architecture in addition to the two planes of the SDN paradigm. As can be seen in their network architecture, the KP is located on top of the control and management planes. The integration of the KP generates behavioral models and reasoning processes for decision making. This architecture enables the KP to have a full view and control of the network through the control and management plane. Other research studies [12,14,29] have a similar architecture to KDN. In [14] the same KP is utilized on top of all the layers, but it uses a cross-layer management and monitoring plane with ML algorithms to manage the rest of the planes. The method proposed in [14] utilizes an ML-based algorithm in both separate orchestration layers and embedded in the management plane. Therefore, it is important to investigate the architecture of KDN in wireless networks to identify the most suitable architecture in terms of flexibility and performance.

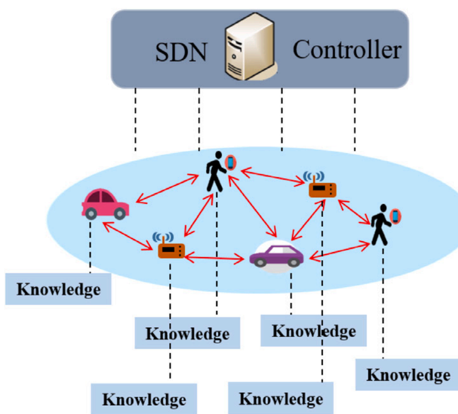
It is predicted that the architecture of the KDN in wire-less networks can be centralized, distributed, or hybrid. In a centralized architecture, as shown in Fig. 6(a), the SDN controller is located at the center of the network and collects information from nodes using OpenFlow (OF), P4, network management, etc. Then, the gathered information is processed through a centralized knowledge plane, and then instructions and rules are transmitted back to each node. The new rules are updated using both direct and indirect approaches. The direct approach uses the previously processed information and sends new strategies immediately back to the user equipment (UE). The indirect approach uses an ML algorithm to determine new rules before being sent to UEs. In the distributed architecture illustrated in 6(b), the individual devices maintain local knowledge. In this architecture, each node collects data from the environment and its surroundings, and then independently applies a greedy-based ML strategy to acquire knowledge. The greedy strategy may be determined from prior knowledge (such as TL) or using new ML-based optimization algorithms. For instance, in a routing scheme, a node can collect information from other nodes and use ML-based approaches to find the best routes and then share the acquired knowledge with other nodes. In the hybrid architecture, as depicted in Fig. 6(c), knowledge is maintained at both the extreme edge and core. Where it is just a matter of keeping the knowledge either updated or synchronized. Both the controller and devices act intelligently together based on the information they collect. This information is processed by ML algorithms to acquire knowledge and injects new rules into the system. The hybrid approach combines the greedy strategy and centralized knowledge to increase the network performance. Further, there can also be a switching strategy to switch between centralized and distributed according to the application.

4. Southbound protocols

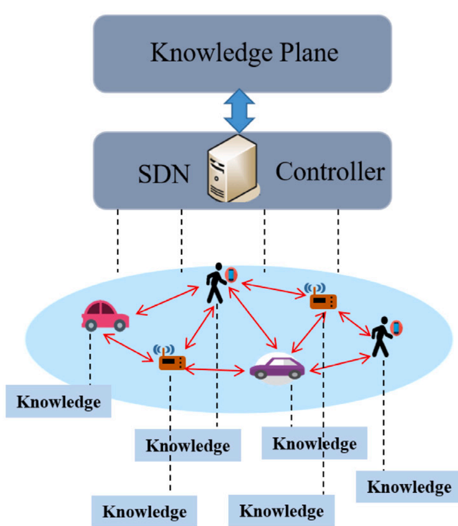
In this section, a brief history of SDN and its enabling standards are presented. Then, complete clarification of P4 against OF and the benefits of P4 over the network are presented. For KDN to be fully functional, data collection is the key, and two powerful tools to do this are OF and P4.Both advantages and disadvantages of these two protocols are stated in this section.



(a) Centralized architecture of KDN.



(b) Distributed architecture of KDN.



(c) Hybrid architecture of KDN.

Fig. 6. Proposed KDN architectures.

4.1. The road to software-defined networking

There are two main problems with the traditional IP network, mainly the complexity and difficulty of network management [30]. The complexity is due to the configuration of individual network devices, such as routers and switches. The difficulty in managing the network is due to the close attachment of the data and control planes. Hence, the distributed nature of the traditional architecture and difficulties in the manual configuration of networking devices have led to the creation of the SDN paradigm [31]. There are different ways of implementing SDN — overlay networking and the use of the OF protocol API to control virtual switches.

Overlay networking was the first deployment of the SDN concept and has been used in data centers before the popularity of the SDN paradigm. In the overlay network architecture, a network layer is added above the basic transport layer (physical layer). Overlay networks use virtual links to create a virtual network on top of an underlying physical platform. A virtual connection between the two nodes at the end of the network is created using tags/labels to create a virtual tunnel (overlay link) from one end to another. The network is programmed to manage the tunnels between the network switches and hypervisors (an intermediate layer between the SDN controller and the associated virtual SDN network). Nodes in overlay networks act based on an overlay topology, which consists of certain behaviors: cooperative or selfish. In cooperative mode, each node creates an overlay link to allow other nodes to route their traffic through different nodes. In the selfish mode, nodes create overlay links to make connections in the network to maximize their benefit. Overlay networks aim to improve QoS by optimum routing decisions. Overlay networks can be deployed by overlay protocol software without the involvement of Internet service providers (ISPs). However, there are some drawbacks to this technique, such as increasing latency, slow-spreading data, and duplicate packets. Moreover, the overlay has no control over routing packets; it only knows the message trajectory before it reaches its destination [32–34].

The development of the SDN paradigm in 2008 and the separation of the control and data planes from the individual network devices solved many previous problems in traditional networks. Furthermore, the programmability functionality of SDN in network simplified network management and enabled innovation [34]. The SDN controller interacts between the switches via the OF protocol API, where OF switches contain flow tables, including rules for handling packets with specific actions [35]. With the arrival of OF, the term SDN was born and used by the research community as early as 2009 [36]. However, it did not have much impact on networking vendors until 2011, when OF eliminated the configuration complexity and automated network management. The development of OF started in 2011, and the latest version was released in 2016. The summary of each version is as follows [37–39]:

- The first OF version was released in March of 2008. However, it was not until December 2009 that OF Version 1.0 reshaped large enterprises and service provider networks.
- Version 1.1 enables innovation by adding multiple flow tables and multiprotocol label switching (MPLS), including modifying packets, complex forwarding actions, and updating an action. This version was introduced in February 2011.
- In December 2011, Version 1.2 added flexibility by authorizing communication between a switch and multiple controllers and supporting IPv6 for the matching process.
- In June 2012, OF Version 1.3 was released, which addressed backbone bridging, per-flow traffic meters, and tunneling.
- Version 1.4 appeared in August 2013 and offered more accessible ways to add new features to the protocol, including type-length-value (TLV) formatting to match fields, role status events, flow monitoring, etc.
- OF Version 1.5 was released in December 2014, enabled fast synchronization among multiple switches and processing and matching ingress packets by adding an egress table. Finally, the latest version of OF Version 1.6 was approved, but it was only available to the open network foundation (ONF) group.

All the versions above have subversions that include bug fixes and minor improvements. However, the major transition in version upgrade occurs between different version volumes (1.x). OF can populate the forwarding tables of switches, and it can also add and delete forwarding entries of almost 50 different header types. Accordingly, the vendors dictate the control plane to which the header they support using the table-type patterns (TTPs) provided by ONF. To manage large-scale networks, SDN operates with OF standards to provide simpler configuration options. While SDN separates the control and data planes, OF only applies a fixed set of protocols to populate the rules in the data plane by using the control plane. However, these protocols are understandable by the fraction of available hardware routers and switches [40]. Therefore, the OF does not control the switch's behavior of the supported protocols. It only provides a way to populate the tables in the switch. Moreover, the current OF has specific protocol headers for forwarding a packet. Forwarding a packer requires forwarding tables, which are known as flow tables in the OF standard. These tables define how a frame is forwarded out of a switch, where the tables operate by matching specific header fields. These tables have grown from 12 to 41 fields in just a few years, presenting a huge challenge. This increased the complexity of the specifications without providing any flexibility by adding new headers. P4 is a tool that reduces the complexity of OF. The necessity of P4 alongside (or operating separately without OF) OF for improving network functionality is promising [5,41]. Fig. 7 depicts the hybrid network utilization of the OF and P4 [40].

4.2. P4

The programming protocol-independent packet processor language is abbreviated as P4. In 2013, the P4 high-level language for programmable protocol-independent packet processors was developed through the collaboration of Barefoot Networks, Intel, Stanford University, Princeton University, Google, and Microsoft [5]. P4 enables the programmability of the data plane and allows switches to process the packet. Hence, vendors and enterprises will be able to develop their own application-oriented software for a programmable switch chip, resulting in several benefits to the network, such as reducing the packet processing time, modifiable packet headers, and switch protocol independence. These programmable switch chips are based on a protocol-independent switch architecture (PISA).

P4 was first introduced in 2014 to address the limitations of the data plane by providing flexibility in programming the data plane in network switches that support OF standards [42,43]. The original P4 language was called $P4_{14}$ which only assumed distinct/specific device capabilities and was able to program a subset of programmable switches [40]. With the evolution of the language, $P4_{16}$ brought new functionalities, such as stable language definition, supporting many switches, and removed the assumption of device capabilities. Generally, P4 processes packets by the pipeline of networking elements, including switches, routers, etc. It is based on a fundamental forwarding model that uses a parsing of the packages and applies match+action table recourses to ingress [41], where the abstract packet forwarding model is shown in Fig. 8. The parser is the process of identifying the headers from each incoming packet. As the header is identified, a lookup function is performed to find the appropriate match according to the header fields, and then applies the action corresponding to the match within the table. The P4 programming language focuses on the specifications of these two procedures and the control flow through the pipelines. These specifications are controlled by programmers who write and execute P4 files. Translation tools are required to execute a file from the P4 program. There are two options for translation: one is by the interpreter on every cycle of execution, and the second is by a compiler once the program is executed. Both approaches possess advantages; while the former has the ability to minimize the error in optimization operations, the latter method can reduce the translation time of each cycle between the development and runtime of the program. The P4 compiler consists of different parts, as depicted in Fig. 9. Some of the most commonly used compilers available to execute a P4 file are summarized as follows:

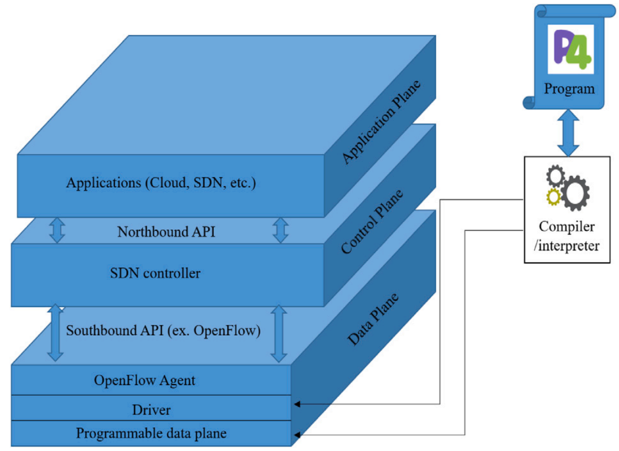


Fig. 7. P4 with OpenFlow architecture.

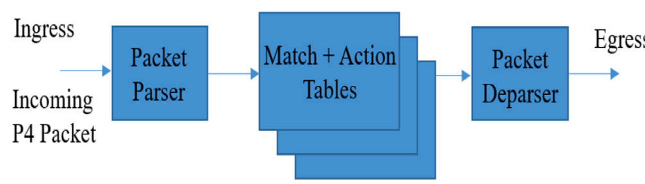


Fig. 8. P4 forwarding model.

(1) *Behavioral Model (BM)*: This is a P4 software switch compiler written in C and C++. The compiler takes a P4 program as an input and then creates a C/C++ program based on the input. The first version of BM is called p4c-behavioral, which depends on a program to generate a high-level intermediate representation of P4 (nicknamed P4-HLIR). P4-HLIR produces a target-independent P4 parser in the Python programming language, which assists the compiler in developing the correct C code for the intended targets. However, this behavioral version has some issues, such as generating extra codes and recompiling codes every time a modification is made to the P4 program. Later, the new version of BM was introduced, BM_{v2}, which is written in C++ and supports static behavior. To run a P4 program on the switch, the P4 source code is first converted to a JavaScript Object Notation (JSON) file, which is then combined with the P4 file and fed as an input to the interpreter. There is a module in BM_{v2} called p4c-bm that generates the JSON configuration and C++ code. The C++ file enables communication between the control plane and data plane on the switch. Finally, the BM_{v2} compiler supports both $P4_{14}$ and $P4_{16}$, and has practical capabilities such as supporting different tools (e.g., control plane, integrated development environment (IDE) software, etc.), multiple back ends, and extensible architecture [44].

(2) *Independent Hardware Target*: The authors in [45] proposed a target-independent compiler so that it can reduce the complexity of implementation. In their technique, the independent target program is linked to a library called the hardware abstraction library (HAL), where this library is implemented for each target. They separate hardware specifications and hardware-independent functionalities to improve portability. However, their method was only implemented for the Intel platform and the Intel data plane development kit (DPDK). In addition, the performance of the compiler is lower than that of hardware-dependent compilers.

(3) *Protocol-Independent Software Switch*: This is a programmable protocol-independent software switch, also known as PISCES.

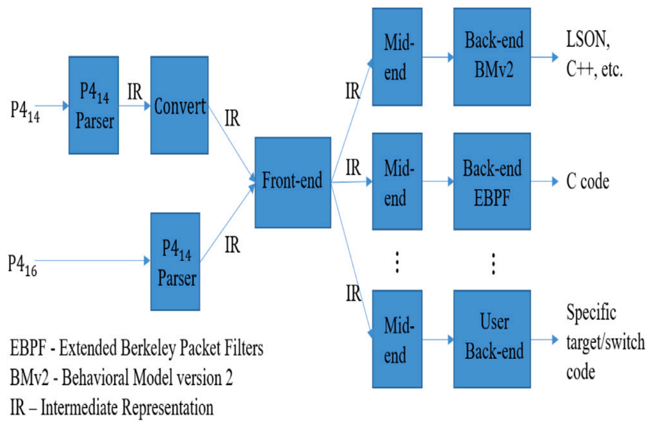


Fig. 9. Inside P4 compiler.

PISCES is a software switch in which forwarding behavior is obtained by high-level DSL or P4. PISCES is derived from Open vSwitch (OVS) and is configured by P4. In this protocol, the programmer must specify how to process the packets. For instance, if in P4 we assign PISCES to process IPv6 packets, the programmer needs to introduce IPv6 packets, including the format and fields of the IPv6 headers. PISCES brings several benefits, such as a personal protocol header, adding/removing a standard header, and easy to add new features [46].

Most of the available P4 programming translators are compilers. However, there exists research for developing interpreters [47], which still needs further attention. Currently, most of the focus is on programmable hardware switches with PISCES that are compiled to a customized software-based POF switch (PVS), where POF stands for protocol oblivious forwarding.

Therefore, P4 enables a new functionality for controlling the forwarding behavior of the switch by populating the tables. Moreover, P4 compilers allow us to use different APIs for switch chips. For most of the current switches in the market, this chip has an on-board invariant programmed module. These APIs are auto-generated by the P4 compilers to populate the switch tables. The new capability to utilize programmable switches will change the way switches operate, but never the less OF is still useful in networking for the old fixed-function switches. Therefore, a program called openflow.p4 has the ability to program switch chips with the support of OF. Hence, P4 and OF can work together for networks such that P4 is the language and OF is the program. Additionally, while OF was essentially designed for SDN networks where the control and data planes were separated, P4 was designed to program the behavior of the switch or the router with no restriction on whether controlled locally by a switch operating system or automatically by an SDN controller. Network information can be obtained using both OF and P4, and can be processed in the KDN to inform the data plan for the new policy. Researchers can choose their own plan toward how and which protocol suits their work, and based on that, they can identify the appropriate translation tools for packet-level programming.

5. Overview of machine learning techniques

Artificial intelligence (AI) is a progressive branch of computer science that deals with automation across various fields, and ML is an application of AI. ML is applied to developing systems to learn from patterns and data without explicitly being programmed [48]. The KDN architecture requires adapting an ML technique to optimize and create intelligence for the network. ML applications have been successfully utilized for network analysis, online customer support, search engines, computer vision, and signal processing applications [49]. As a result, research studies based on ML on various aspects enable the KDN paradigm to access and adapt a suitable ML algorithm for the appropriate task. In this section, we survey a number of KDN-relevant ML methods, including SL, UL, RL, NNs, and TL. We briefly explain each algorithm and explore different techniques before using them in Section 6. Therefore, anyone interested in the field will acquire abstract knowledge of ML-based techniques and understand why and how an ML algorithm can be utilized in wireless communication applications.

5.1. Supervised learning

Supervised learning (SL) is a basic ML algorithm that takes an input feature network x and tries to produce a model from the output information y. SL is further divided into two categories: regression and classification. In regression problems, we try to predict continuous value output, and in classification problems, the prediction is for discrete value output. Classification problems are used to distinguish between different things, such as prediction in image processing, to differentiate between a cat and a dog [50]. In contrast, regression problems cannot be considered as a classification problem [51]. The following are the popular survey works on SL techniques:

(1) *Linear Regression*: Linear models are provided for prediction as a weighted sum of feature inputs. Linear regression uses gradient descent to minimize the loss function and optimize the system performance. Linear regression can only represent linear relationships, which reduces the prediction performance, but delivers a fast and acceptable result for many applications. Further specifications and information regarding linear regression are available in [52].
(2) *Logistic Regression*: Classification problems are used for logistic regression to assign observations to a discrete set of input classes. For example, the common use of this algorithm is to predict spam emails, student pass or fail grades, fraudulent websites, and so on. This learning algorithm uses the gradient descent method to optimize the solution. In contrast to linear regression, in the logistic regression model, a number between 0 and 1 is assigned to each data point instead of fitting a straight line or hyperplane. The predicted values are the probabilities generated by the sigmoid function. In other words, from the function outputs, the learning algorithm decides which category the output data belongs. This algorithm has some drawbacks that can also be applied to linear regression. For instance, if the data are perfectly separated, the algorithm can no longer be trained, or interpretation is more difficult because the interpretation uses multiplication rather than addition. However, some advantages make the algorithm interesting, including multiclass classification and the probability distribution of the data. More details about the logistic regression are found in [52].
(3) *Support Vector Machine (SVM)*: This method is a linear classifier that separates the training data that are n-dimensional vector by using an $n-1$ hyperplane. The optimum hyperplane must be selected to increase the marginal space between the two given classes. Furthermore, this algorithm works for non-linearly separable (non-linear classifier) datasets using kernel functions, including Gaussian and polynomial kernels. Although SVM requires a long time to distinguish the alarmed dataset and degrade as the noise increases in the dataset, this method works effectively with high-dimensional spaces. This method uses a fraction of the memory and tightly margins the separation [53, 54].
(4) *K-Nearest Neighbors (KNN)*: The KNN algorithm is a straightforward and valuable algorithm with no explicit training requirements. This method is used for both classification and regression problems, where the input data consist of the K closest training

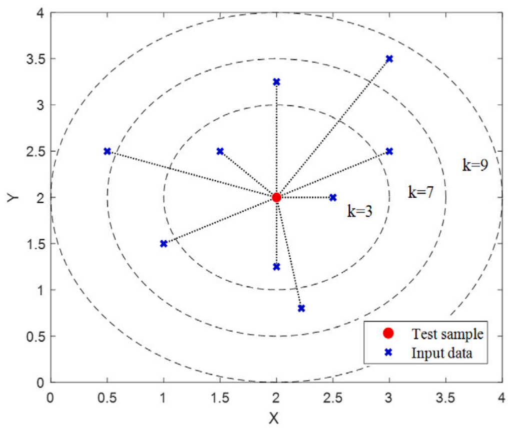


Fig. 10. An abstract model of the KNN algorithm.

dataset in the feature space. The K value is the hyperparameter and should be selected carefully, but the optimal K-value can be chosen after running the KNN several times with different K-values. The resulting KNN outputs can be used for both classification and regression. In classification, the basic principle is to decide the category of a test point based on the decision made by the majority votes of K nearest neighbors. In the regression, the test point is categorized in a class by the average values of its K nearest neighbors. Additionally, this method can be adapted by assigning weights to neighbors such that the nearest points are more involved than the distant points. The KNN method is illustrated in Fig. 10. A more detailed explanation of KNN can be obtained in [55].

(5) *Decision Tree*: One of the most common ML algorithms is DT, which is a statistical model for classification problems. This ML technique is used to classify data and formulate a dataset in a hierarchical structure, such as a flowchart representation. This flowchart is usually a tree-based structure, and the algorithm starts from the root and classifies the dataset until it reaches the leaf. The representation of one class DT is as follows: Each split of the tree is based on a condition on a particular attribute, and leaves are the classes. For a simple example of DT, the iterative Dichotomiser 3 (ID3) is explained as follows:

(i) First, it assigns the original dataset (D) to the root node.
(ii) Iteratively calculate the entropy $H(D)$ or the information gain $IG(D)$ of every attribute of set D.
(iii) Then, it partitions set D into subsets using the smallest resulting entropy from step ii.
(iv) Making a DT node consisting of that attribute.
(v) Recurse on D using the remaining attributes.

The DT method works for both regression and classification problems with many advantages, such as ease of interpretation, requires fewer data points to learn, and works well with a large dataset. However, minor changes to the large dataset require a new training sequence, and overfitting is the algorithm's limitation. For implementation and further information, refer to [55]. Fig. 11 provides a simple example of a decision tree.

5.2. Unsupervised learning

In unsupervised learning (UL), the input data are unlabeled data, where the algorithm has to find patterns and hidden structures to learn a useful function. The enormous data collection by devices and sensors results in a lack of labeling due to the unavailability of funds to pay for manual allocation or the nature of the data itself. The UL is extensively used in clustering and data aggregation [56]. The following algorithms are the most common UL techniques:

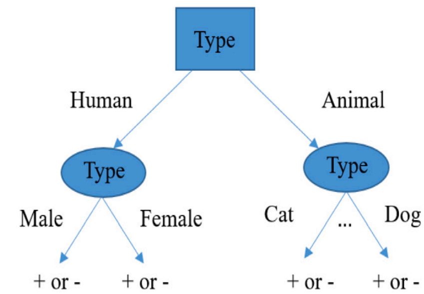


Fig. 11. An illustration of decision tree model to find out whether a person is overweighted.

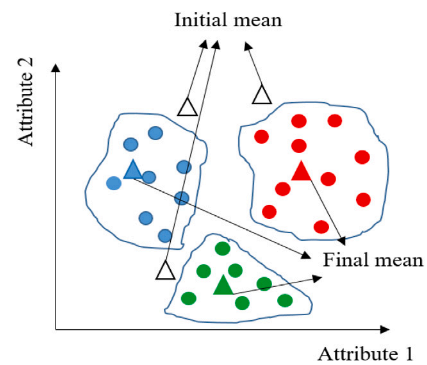


Fig. 12. An abstract view of K-mean clustering.

(1) *Clustering Algorithms*: In clustering algorithms, K-mean clustering is widely used to segregate n unlabeled data points into K clusters. To initiate this algorithm, only the dataset and the desired number of clusters are required. The most popular version of this method is iterative refinement, where K is randomly chosen, and after every iteration, the algorithm corrects itself [55]. The difference between this algorithm and KNN is that in every iteration, every point is assigned to a cluster using the Euclidean distance (closest to the centroid). In each iteration, the algorithm learns and updates the centroids of each cluster. The algorithm operates until the end of the iterations, or when the centroids of the clusters do not change significantly. A basic observation of this algorithm is shown in Fig. 12.

(2) *Principal Component Analysis (PCA)*: This is a statistical procedure that transforms a set of correlated variables into a set of linearly uncorrelated variables using orthogonal transformations [55]. The fundamental idea of PCA is to reduce the dimensionality of the data and optimize it to major components or features. This statistical method represents the datasets in more economical and smaller observed variables for faster data processing in ML. Moreover, PCA works optimally with linear models for feature extraction, data compression, and redundancy of variables, such as image processing, signal processing, communications, and control systems/theory [57]. In this algorithm, singular value decomposition (SVD) plays an essential role in computing lower-dimensional data. SVD extracts the eigenvalues from the covariance matrix, which is the best approximation of the original dataset with fewer arguments. For more information on computation and simulation, refer to [55,58].

5.3. Reinforcement learning

Reinforcement learning is a popular ML techniques that through interaction with the environment, the RL agent learns from trial and error using consequences of its experiences and actions [59]. Some of the commonly used RL-based algorithms are described below.

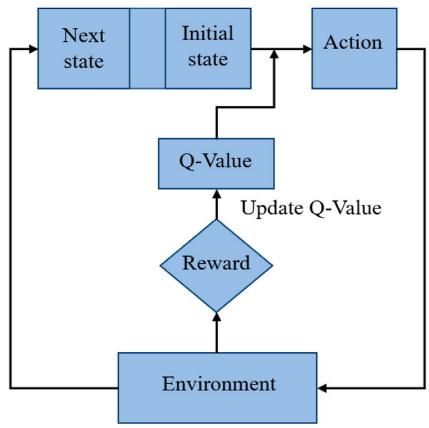


Fig. 13. An abstract view of Q-learning method, where every action taken in any state will be observed by the environment and generates a reward to provide insights about how good was the agent's action.

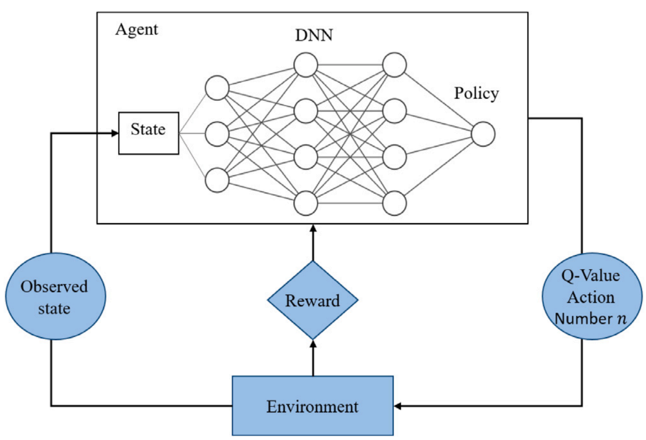


Fig. 14. The procedure of deep reinforcement learning.

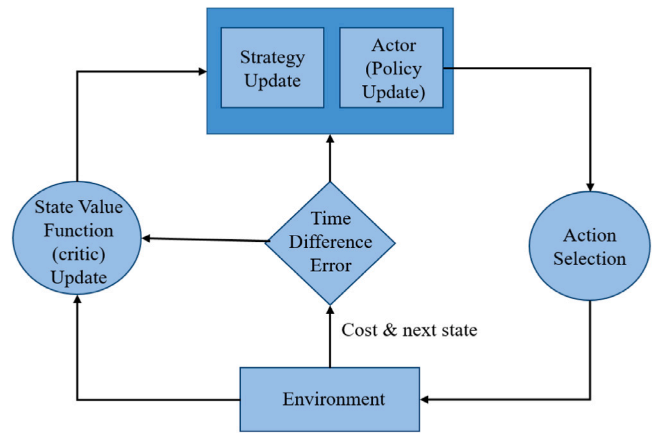


Fig. 15. An illustration of actor–critic learning algorithm.

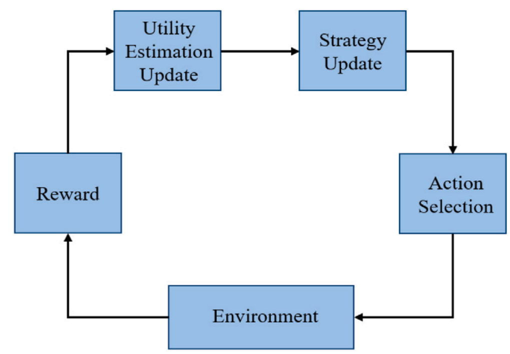


Fig. 16. Joint utility and strategy estimation-based learning process.

(1) *Q-learning*: Q-learning is the most popular RL technique that uses Q-value and Q-function to find optimum action policies. Specifically, the agent interacts with the current given environment to continuously learn the Q-values and maximize this value. This algorithm starts with an initial state and an action, followed by the epsilon-greedy policy. For each performed action, the Q-value is learned through the optimal (greedy) policy, enabling the agent to take any action based on the largest Q-value under the current state. Details of the Q-learning method can be found in [26,60]. Moreover, fuzzy Q-learning is used to deal with continuous state spaces with a certain number of given rules [61]. A flowchart of the Q-learning method is presented in Fig. 13.

(2) *Deep Reinforcement Learning (DRL)*: Deep RL is a combination of deep learning and RL, which correlates the value function with corresponding actions and states [62]. DRL uses these two principles to approximate the optimal Q-values. Moreover, DRL techniques facilitate an NN with an RL architecture to enable agents to learn the best action in a specific environment. DRL is mostly used in complex decision-making tasks with unstructured environments and can handle large datasets. Recently, DRL has made great strides in vehicle-to-vehicle (V2V) communication [63], wireless communications [64], and video games [65,66]. More details on DRL can be found in [67] for the readers. An overview of this concept is shown in Fig. 14.

(3) *Actor–Critic Learning*: There are two main types of RL: value-based and policy-based. Value-based attempts to approximate the optimal value by mapping between the action and value to find the optimal policy without the Q-value. At the same time, policy-based policies such as policy gradients rely upon optimizing parametrized policies. To take advantage of both types, these two were merged and established actor–critic learning, which means that one will compute an action based on the current state and the other approximates the Q-value of the action [68]. An actor–critic learning method comprises an actor for error correction, a critic for interaction, and an environment where the actor interacts. As a result, the actor selects an action based on the current policy, producing a cost metric. Subsequently, the critic updates the state value function according to the time-difference error. Finally, the actor updates the policy. Each action is reviewed a finite number of times for each state such that the algorithm converges to optimal values. Additionally, the strategy of the system is updated according to the learned policies using the Boltzmann distribution [69]. The basic principles and procedures are shown in Fig. 15.

(4) *Joint Utility and Strategy Estimation Based Learning*: The utility and strategy estimation-based learning relies on the same concept as the classical RL. However, the main difference is that the agent receives an estimation of the expected utility from the environment and the updated reward. The probability distributions are modified based on the utility estimation, where the probabilities (also known as strategies) are selected actions [70,71]. Using this algorithm, the regret of each action in the process can be obtained using the received reward and utility parameters. Then, regret can be used to update the strategy. The main advantage is that the algorithm can be fully distributed

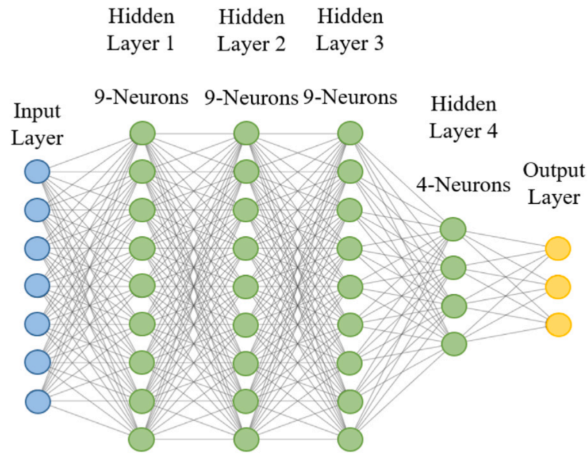


Fig. 17. The architecture of DNN.

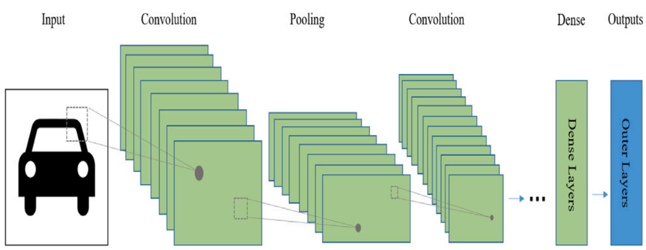


Fig. 18. An overview of training process of a convolutional neural network architecture.

when the reward is immediately calculated locally; for instance, the transmission rate between the transmitter and receiver. This algorithm is often referred to as an equilibrium concept in game theory in many survey papers, such as coarse correlated equilibrium and logit equilibrium. The concept of this algorithm is illustrated in Fig. 16.

5.4. Neural network

A neural network is a type of artificial intelligence for information processing that imitates the human brain. The neural network structure consists of thousands of closely connected, simple processing nodes. Neural networks are organized into layers, and in each layer, many nodes move the data. With the development of graphics processing units (GPUs) to accelerate the processing time, NN has attracted considerable attention from researchers and companies [72]. The following NN techniques are explained in this study.

(1) *Deep Neural Network (DNN)*: As shown in Fig. 17, the components of the structure are densely connected by the neurons in the network layer. Each neuron in a layer is connected to the rest of the neurons, resulting in the structure of the DNN. Each neuron corresponds to a weight for the input and an activation function for the output. The input data are transformed from layer to layer, with no direct connection between the two non-consecutive network layers. The main advantage of a DNN is that it can automatically deduct and tune the features to obtain the desired output. For the optimization of network parameters, DNN uses backpropagation (one of the most popular learning techniques for multilayer neural networks) and various gradient descent algorithms, such as Adam and Momentum [73].

(2) *Convolutional Neural Network (CNN)*: The convolutional neural network is a class of artificial NN that was developed during the 1980s [74]. A CNN is designed to adaptively and autonomously learn spatial hierarchies of patterns by using different building blocks, including convolution layers, pooling layers, and fully connected layers. Among them, convolutional and pooling operations are the two main building blocks in CNNs, which are feature extraction [75]. Using these two layers enables the CNN to solve relatively complex models based on the progressive nature of the structure, as shown in Fig. 18.

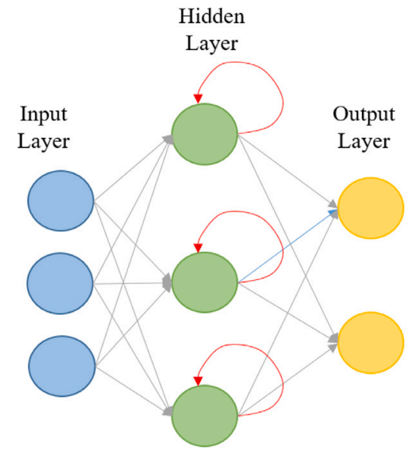


Fig. 19. The architecture of RNN.

(3) *Recurrent Neural Network (RNN)*: The principle of RNN is to save the previous output and feed it back to the input while having hidden states to assist the algorithm in predicting the outcome of the layer [76]. Additionally, there are connections in the hidden layers of the RNN architecture, where all the inputs, including the current and former inputs, impact the output. Hence, an RNN has the ability to remember [26]. Fig. 19 shows the hidden layers and connections of the RNN architecture.

(4) *Autoencoder*: The autoencoder aims to model a set of data to learn and approximate the system function. The autoencoder has other variations, such as sparse autoencoder and denoising autoencoder. The autoencoder also applies backpropagation and sets the target output equal to the input. The components and process of the autoencoder are shown in Fig. 20; after the autoencoder is fully trained, the decoder is removed from the network, and only the encoder is kept as a feature extractor. This powerful NN can learn a robust representation of the input with a limited number of neurons such that the network can construct the input at the output. However, the autoencoder has some disadvantages that worth mentioning. First, an autoencoder requires an identical distribution of the training data at its input before it can work properly. Second, the number of neurons in the hidden layer has a direct impact on its performance [26].

(5) *Extreme Learning Machine (ELM)*: The extreme learning machine is an NN with multiple hidden layers with randomly chosen parameters, often referred to as a feed-forward neural network (FFNN). This is the simplest and the first artificial neural network, which aims to approximate a function to map an input to an output such as the XOR function. Different layers connect each node to other nodes in the network, including the hidden layer, input, and output layers. The information flows only in one direction, and it does not include any feedback, or the nodes do not form a cycle, differentiating it from the RNN.

5.5. Transfer learning

Transfer learning is the process of avoiding learning from scratch and utilizing knowledge from a specific task to the target domain to

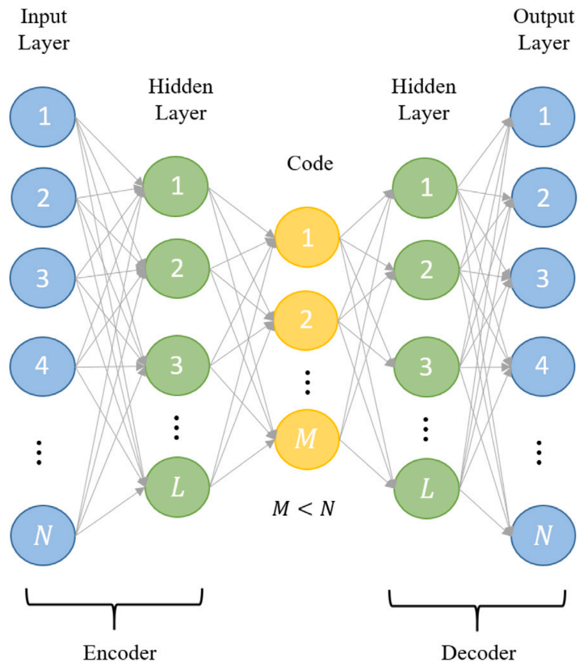


Fig. 20. Illustration of the structure of an autoencoder.

assist learning more robustly. For deep learning and RL, knowledge can be defined as weights and Q-values, respectively. For instance, when deep learning is adapted for resource management, the ML process can use the weights that have been trained for other resource management tasks as the initial weight. For example, a similar network with the same number of nodes and similar behavior with a trained ML and a fully operational resource management policy can provide knowledge for other similar networks. Moreover, in RL, the Q-values learned from an environment can be used in a similar environment to make better decisions during the initial stage of learning. The specifications for utilizing TL with RL can be further studied in [77]. While there are advantages of having prior knowledge for learning a pattern, there are some limitations and negative impacts on the performance, which needs further attention that exceeds the scope of this study.

Most of the above ML techniques can be deployed in several network applications for automation and optimization. ML algorithms provide information and knowledge for different tasks. In the next section, the application of ML in KDN for wireless networks is discussed. First, a general introduction to each application was explored. Second, the important characteristics of each surveyed study with supporting ML algorithms were investigated. Finally, a possible KDN architecture that can be adapted for this specific study is presented. This study was expanded over various parts of a wireless network, including resource management, networking, mobility management, and localization. Therefore, it is crucial to identify the studies within these parts that can be potential use cases in generating knowledge in the KDN paradigm. These studies are the building blocks for achieving a fully knowledge-based network in future wireless networks. Table 2 presents the selected number of surveys with different applications supporting various parts of wireless networks.

6. Application of machine learning for knowledge defined networking

Owing to the exponential growth of data traffic, wireless networks will require advanced technical solutions in the near future. As a result, the traffic load among BSs, the complexity of wireless channels, the emergence of self-driving vehicles, device-to-device (D2D), and machine-to-machine (M2M) communication requirements make traditional networking intolerable. To overcome such issues, key technological advances in the network, such as the KDN, can support and potentially solve our problems. In the following subsections, four major parts of the communication stack are reviewed based on previous research.

(1) Resource management applications based on spectrum utilization, power allocation, QoS, BS switching, cache, and backhaul management were investigated.
(2) Networking applications, such as routing, clustering, user/BS association, and data or traffic aggregation have been explored.
(3) Mobility issues, such as user mobility prediction and handover management, were investigated.
(4) The importance of indoor localization is presented.

These four criteria are the essential parts of each network, and they need optimization to keep up with standards and demands in future heterogeneous networks. In this section, we discuss recent ML studies in a wireless network. Here, ML-based information is stored as knowledge for that specific network to facilitate other similar networks.

6.1. Resource management

Nowadays, resources among networks are scarce and expensive. Many studies have started to use optimization methods and introduced new ideas for resource management [78]. However, recent studies have utilized ML algorithms to improve the efficiency. These studies focused on spectrum allocation, power management, QoS, BS switching, cache, and backhaul management. Knowledge that KDN provides is beneficial to these standard problems in wireless networks. Once the complexity of the network increases in 6G, there needs to be a centralized intelligence that can receive general network information and process that information through an ML algorithm to produce meaningful knowledge. Therefore, it is an excellent opportunity to study research works and identify those with the potential to attain useful knowledge in the KDN framework.

6.1.1. Spectrum allocation

As the number of devices increases in the network, such as IoT, cellular, and sensors, spectrum shortages have drawn significant attention. The increase in data traffic now forces efficient spectrum allocation and management strategies to enhance scalability and improve QoS. In the following section, RL is investigated for possible knowledge extraction for spectrum management when using the KDN framework in the network. Table 3 summarizes the studies surveyed in this section.

Knowledge derived from reinforcement learning: In [79], spectrum allocation in cognitive radio networks (CRN) is presented, and the non-dominated sorting genetic algorithm-II (NSGA-II) is used as a method to combine the evolutionary algorithm and RL method. As a result, they proposed NSGA-RL for self-tuning and spectrum allocation in an efficient manner. They used the RL algorithm to learn the value of the control parameter during the training phase. Their method evaluates the initial values of the Q-table and updates the Q-values iteratively to obtain optimal values. Based on their algorithm after network initialization, non-dimensional sorting via tournament selection, crossover, and mutation is calculated, representing offspring population generation. They use population as knowledge to find the near-optimal and self-tuning control parameter to allocate spectrum-based and increase the overall network capacity. This algorithm is suitable for the centralized architecture of the KDN, where a multi-objective optimization problem for spectrum allocation is modeled based on the RL algorithm.

In [80], the authors developed a dynamic spectrum allocation algorithm in a distributed manner using a deep multi-user RL. Their method allocates the shared bandwidth into orthogonal channels, and users access the spectrum at each time slot based on a random-access protocol. First, users attempt to transmit packets with a certain probability. Then,

Table 2
The application of machine learning methods on different areas of telecommunication.

	Applications	Recourse Management						Network				Mobility		Localization
	ML Methods	Spectrum	Power	QOS	BS Switching	Cache	Backhual	Routing	Clustering	User Association	Traffic Classification & Data Aggregation	User Mobility Prediction	Handover Management	Indoor Localization
Supervised Learning	Supervised learning with /without NN		[89]			[110], [113]–[115]		[137], [142]–[144]	[164]	[170], [171]	[181]	[188]	[193]–[196]	[218]
	linear regression							[140]						
	Logistic regression								[161]					
	Support Vector Machine					[115]			[162]					[214]–[216]
	K-Nearest Neighbors								[163]					[210], [217]
	Decision tree			[100]				[141], [142]						
Unsupervised Learning	Unsupervised learning with /without NN							[145], [149]	[168]			[189], [190]		[219], [222]–[224]
	Clustering Algorithm							[146]	[165]–[167]					
	Principle Component Analysis (PCA)							[145]	[168]					[220], [221]
Reinforcement Learning	Q-Learning	[79], [81]–[83]	[95], [96], [98]		[103], [106]	[117]	[134], [135]	[151], [153]–[156]		[172], [173]			[199], [205]–[207]	
	Fuzzy Q-Learning				[107]					[174]			[198], [200], [204]	
	Deep Reinforcement Learning	[80]	[92]		[105]	[121]–[123], [126]		[150], [159], [160]		[175], [176], [180]		[191]	[201], [208], [209]	
	Actor-Critic Learning		[97]		[104]			[160]						
	Joint Utility and Strategy estimation based learning	[84], [85]	[99]			[116]	[133]							
Neural Network	Conventional Neural Network					[115]		[143]	[168]					
	Recurrent Neural Network	[86]				[113], [114], [121]							[195], [196]	
	Deep Neural Network		[89], [92], [97]	[101]				[144]	[164]				[194]	
	Autoencoder													[219], [224]
	Extreme machine learning					[110]								
Others	Transfer Learning					[127], [128], [130]					[182]			

each user receives an acknowledgment of whether the packet has been received. Subsequently, each user maps its current state to an action based on the spectrum access obtained by a trained deep Q network. Their algorithm proves that users can learn the best policy based on the objective network utility. The proposed algorithm provides twice the channel throughput compared to slotted-aloha with optimal probability, which requires full knowledge of the number of users. Hence, this suggests that the output of the Q-learning algorithm in their study can be used as knowledge for KP in wireless networks, where the number of nodes is not visible by the controller, and the controller needs to perform the best action and allocate an efficient channel to users. The proposed algorithm suggested a distributed adaptation of transmission parameters, where the knowledge obtained from a network by users can be used in a distributed manner in the KDN architecture.

As data traffic increases in mobile networks and fixed spectrum allocation of operators becomes a major problem, inter-operator spectrum sharing has been proposed as a solution. This solution brings benefits but also introduces new challenges. The authors of [81] introduced a new paradigm called inter-operator proximal spectrum sharing (IOPSS) to intelligently offload users from an overloaded BS to the neighboring BSs based on the spectral proximity. A Q-learning framework is equipped with each BS to dynamically determine the network load and efficiently utilize its spectrum in a self-organizing manner. The state of the system depends on the network load experienced by the BS and is defined as a discretized value. At the same time, the action of each state is determined by the tuple of spectral sharing parameters associated with each neighboring BS. The spectral sharing parameter for a BS includes the required spectrum resources (where spectral resources in this study are considered as resource blocks (RBs)) from a neighboring BS, the probability of a user that the neighboring BS can support with the strongest value of the signal-to-interference-plus-noise ratio (SINR), and the fraction of RBs that need to be reserved as requested. Simulation results demonstrate that the IOPSS-based Q-learning algorithm can intelligently offload users from congested BSs and help cellular operators to enhance the user's quality of experience (QoE) and efficiently share spectrum resources. In this study, the IOPSS Q-learning framework was executed in a distributed manner at every BS. This framework is suitable for both centralized and distributed KDN architectures. If the network is seen based on every cluster, where all the BSs in a network are assumed to be a greedy-based BS, then a distributed KDN using the proposed work will maximize each BS's resources. However, this method can also be assumed as a centralized algorithm for the master controller that manages all the BSs, which eventually suggests a hybrid structure of KDN using the output data resulting from the proposed study.

In [82], the authors proposed a dynamic spectrum allocation strategy for maximizing throughput in millimeter-wave (mmWave) ultra-dense networks. They proposed a temporal–spatial spectrum-reuse scheme to improve spectrum management. Similar to other works, spectrum management is formulated using a non-cooperative game, which ensures Nash equilibrium (NE). Using this technique, secondary users (SUs) are adapted with a distributed Q-learning algorithm to interact with the environment and achieve NE without any global information exchange between SUs. The actions and rewards of the Q-learning algorithm are based on the system's channel selection and channel rate, respectively. In contrast to other Q-learning methods, where the state–action pairs will result in a Q-value, the action only defines the Q-value. The player's Q-value is updated in each iteration based on the weighted sum of the current Q-value and the achieved reward. From their comprehensive analysis, their learning algorithm speeds up the convergence procedure and helps users with minimum access latency. Similarly, a heuristic channel allocation policy focusing on temporal–spatial spectrum reuse with the help of multi-armed bandit (MAB) theory is presented [83]. They incorporated a three-stage policy to improve the computational cost of a centralized channel allocation policy. The algorithm uses a distributed greedy graph coloring method to determine the optimal channel access ranks of SUs during the first two stages. In the third stage, based on the MAB

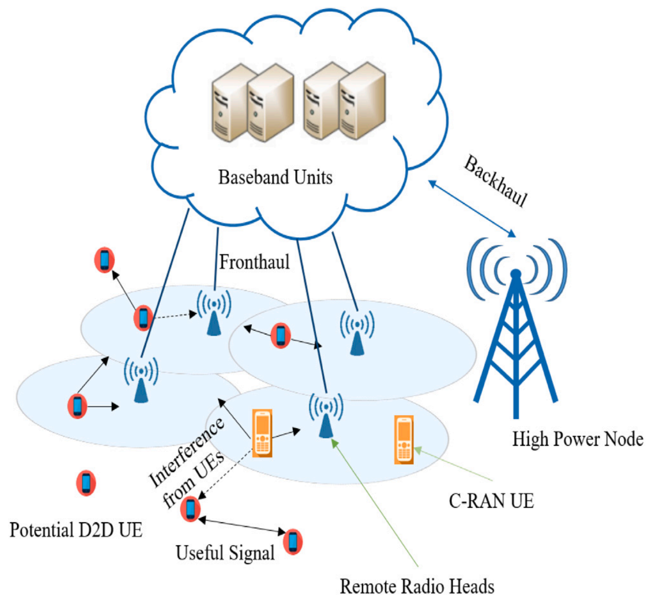


Fig. 21. An abstract system model of D2D enabled C-RAN.

theory, optimal channel allocation is obtained. The simulation results suggest that the proposed policy outperforms other policies in the literature based on temporal–spatial spectrum reuse. The study in [82] can assist in knowledge extraction and maximize the network benefits for a distributed KDN architecture. In contrast, in [83] the spectrum allocation problem is formulated into centralized and distributed parts, which can best fit our hybrid architecture in KDN.

To improve resource management, D2D communication was introduced to decrease the load on the BS. In [84], the authors proposed a distributed approach for resource allocation and communication mode selection for potential D2D pairs using a joint utility and strategy estimation algorithm. The proposed system model is shown in Fig. 21 consisting of potential D2D pairs, BS, C-RAN, and user equipment (UE) that are unable to perform D2D communication. They investigated D2D-enabled C-RANs to improve the spectral efficiency using RL techniques. The action of the learning algorithm consists of the communication mode selection and subchannel allocation for each D2D pair. After each pair selects its action, the remote radio head (RRH) association and power control of the D2D pairs is modeled. Then, based on the reinforcement algorithm, the system obtains the instantaneous utility (known as the received spectral efficiency), which is updated for each action continuously. This study aims to enable D2D pairs to self-optimize their resource allocation and perform mode selection under different practical constraints, including fronthaul capacity and inter-tier interference constraints. The numerical simulations demonstrated near-optimal performance and better spectral efficiency. From the proposed method, it is evident that this study best suits the distributed architecture of the KDN, where D2D pairs will maximize their resources using the proposed algorithm.

In heterogeneous networks, the authors of [85] proposed a multi-objective fully distributed strategy based on RL for self-configuration and optimization in LTE small cells. The proposed algorithm uses a joint utility and strategy estimation under QoS constraints to minimize the received intra- and inter-tier interference for femtocells (FCs) and reduce inter-tier interference from FCs to eNBs. Their algorithm utilizes RL techniques to simplify their problem formulation compared to works where global knowledge and complete CSI are unavailable. Hence, we can utilize the RL ability to gather information from the interaction between BSs and users. They identified two sequential learning levels. During the first phase of learning, available spectrum resources were selected, while resource optimization was performed during the second phase. Different utilities are defined for each level, namely the spectrum modeling utility (SMU) and spectrum selection utility (SSU) for levels one and two, respectively. To make the learning more accurate, FCs use three different learning algorithms, including the modified Bush and Mosteller (MBM), modified Roth–Erev (MRE), and gradient follower (GF) learning algorithms. During each learning process, each FC chooses one learning strategy based on its objectives and capabilities. The algorithm can self-organize the available resources and tune the SBS parameters based on interference and QoS. The simulation results prove the higher cell throughput and significant reduction in inter-and intra-tier interferences. A thorough study was conducted for spectrum allocation, user association, and load balancing for heterogeneous small-cell networks (SCNs) in [86]. An echo state network (ESN) framework is utilized with RNN [87,88] to optimize resource allocation with uplink and downlink decoupling in SBS-LTE-U systems. They formulated a non-cooperative gaming system where the SBSs were the players with the goal of finding the optimal utility function. Every SBS seeks to attain optimal spectrum allocation to enhance the utility function. The utility function must maximize the downlink and uplink sum rates while managing the traffic load in both the licensed and unlicensed bands. Further, based on ESN, resources can be scheduled, and the optimal utility function can be established with minimum knowledge of all users. The algorithm enables BSs to autonomously decide on a spectrum allocation strategy with the maximum throughput of each user. Their simulation results showed a performance improvement of 167% compared to the Q-learning algorithm. In [85] a distributed self-organized spectrum allocation scheme was proposed for FCs. The proposed method is suitable for both distributed and hybrid schemes in the KDN. Similarly, the method in [86] can also be adapted for both distributed and hybrid architectures, in which the hybrid scheme, the centralized SBSs use this technique to act greedy for load balancing and efficient spectrum reuse.

6.1.2. Power management

Power management is a key feature of wireless networks. One of the biggest concerns is that as the number of devices in the network increases, each node will eventually demand a higher data rate, which forces the transmitter to increase its power. However, increasing power increases the interference between the devices. Therefore, the development of a suboptimal power allocation algorithm is necessary for future wireless networks. Exploiting knowledge from ML algorithms enables us to deploy the KDN architecture in 5G and beyond. In power management problems, knowledge is essential to decrease both the interference and energy consumption. In the following sections, supervised and reinforcement learning algorithms are explored to examine how knowledge can be extracted and deployed in the KDN. Table 4 summarizes the studies surveyed in this section.

(i) *Knowledge derived from SL*: In [89], the authors investigated the problems of optimization algorithms, which often lack adequate performance in real-time processing and suffer from complexity. They proposed a centralized SL algorithm to train a deep neural network (DNN). In particular, their algorithm is compared to the weighted minimum mean square error (WMMSE) optimization algorithm [90], which achieves 90% or higher efficiency. The WMMSE is used in interference management and requires complex matrix operations and bisection in each cycle [90,91]. Further, they used a fully connected NN with channel coefficients at the input layer, multiple hidden layers, and an output layer. The system output is the optimal allocated power value. As a result, their algorithm can be adapted for power allocation in wireless networks and utilized as centralized knowledge to obtain efficient power values and mitigate interference. Having the knowledge of the DNN at the KP layer empowers the system to operate at its best and perform optimization simultaneously.

Table 3
Knowledge-based strategies for spectrum management.

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[79]	Utilizing optimal values of Q-table for similar CRNs	Centralized	Q-learning	Self-tuning spectrum allocation, near-optimal solution and improving network capacity	Handles high computational complexity, however performance is highly dependent on the control parameters
[80]	Providing a general real-world solution for spectrum access using a trained multi-user DQN	Distributed	Deep reinforcement learning	Better channel throughput compared to slotted Aloha and optimal spectrum allocation	Offline learning and online real-time spectrum access, decreasing routing overhead but subjected to time constraint for mobile users
[81]	Intelligent user offloading and self-organizing spectrum allocation	Hybrid	Distributed Q-learning	Maximizing the user's QoE	Efficiently share the spectrum between BSs to offload users but challenges of user mobility have not been studied yet
[82]	Self-adaptation of SUs based on trained Q-learning algorithm	Distributed	Distributed Q-learning	Achieving NE	Maximizing transmission efficiency while coordinating interference and suitable for 5G communications but include redundant information
[83]	Efficient channel reuse for SUs once the optimal policy is learned	Hybrid	Multi-armed Bandit techniques	Enhancing spectrum utilization with temporal–spatial spectrum reuse	The channel allocation policy has fewer regrets than other policies, but it works work static networks
[84]	Using the built-in strategy profile for intelligent communication by using mode selection (referred to as selecting D2D or C-RANs data transmission) and resource allocation of D2D pairs	Distributed	Joint utility and strategy estimation based learning	optimizing spectral efficiency	D2D pairs can self-optimize their mode selection and resource allocation, although it is time consuming
[85]	Autonomously identifying and optimizing spectrum usage in femtocells using the learning strategy	Distributed or Hybrid	Joint utility and strategy estimation based learning	Minimizing the intra/inter-tier interference and increasing the throughput	Reduces waste of memory and the unnecessary information
[86]	Self-organizing framework to optimize resource allocation in both uplink and downlink LTE networks	Distributed or Hybrid	Multi-agent reinforcement learning based on echo state networks	Load balancing and efficient spectrum allocation	Choosing the optimal resources by SBS given minimum information about users and network

(ii) *Knowledge derived from reinforcement learning*: Research in [92] introduced a deep Q-learning algorithm for dynamic power allocation based on collected channel state information (CSI) and QoS. Their proposed distributed algorithm is based on the model described in [93], which does not rely on the network size, and it uses a robust technique for the dynamic changes of the network. They considered a single frequency band for transmission with synchronized time slots. A classical power allocation method is utilized in the initial stage of the network. Then, an RL agent interacts with the environment and learns by observing the rewards by trial and error over time [59,94]. To optimize the system and mitigate the problems with Q-learning, a deep Q-network (DQN) is used to estimate the Q-function. Their algorithm showed a fast and suboptimal power-allocation technique for various dynamic wireless networks. This will have a massive opportunity in the future for heterogeneous networks to be implemented and used as prior knowledge and facilitate the users such that optimum power is allocated to all the users quickly and efficiently. This method is more reliable than the other methods because of its response to dynamic changes in the network. A distributed knowledge plan can benefit from this algorithm and utilize the information to optimize power when the network experiences abrupt changes.

For heterogeneous networks, the authors of [95] proposed a decentralized solution for power management and cell association using RL techniques. They have focused on inter-cell interference coordination (ICIC) using both time-domain and frequency-domain, where macro-cells and pico-cells autonomously learn to optimize their transmission power. In the time domain, the learning agent on pico-BS (PBS) intelligently self-organizes the cell range to effectively offload traffic from the macro-BS (MBS). The action performed by the two agents on both BSs is range expansion and power adjustments on each resource block, where PBS performs range expansion and power adjustment by MBS. The future state of each agent depends on the SINR condition of each UE in the network, while the received cost is the target value of the SINR to serve each UE to meet the total transmit power constraint. In each iteration of learning, PBS chooses an action resulting in the smallest Q-value, and based on this, MBS chooses its action. In the frequency-domain scenario, the action of Q-learning differed from the time domain. It uses single and multi-flow carrier aggregation in which a single UE can be served simultaneously by different BSs over different tiers/layers (with two different carrier frequencies). Therefore, using Q-learning will enable PBS and MBS to self-optimize system performance using ICIC. Moreover, a fully automated multi-agent Q-learning technique was developed to facilitate heterogeneous cellular networks and to model the channel and power levels of D2D pairs [96]. Each pair attempts to maximize the value obtained by the difference between the throughput and power consumption cost, which is achieved via the defined SINR constraint. The proposed model is formulated using a stochastic non-cooperative game, where each pair of devices becomes a learning agent to learn the best policy from locally observed information. Their simulation results showed an acceptable convergence rate and near-optimal performance with a few learning iterations. The study proposed in [95] suggests a distributed policy for KDN to incorporate self-healing, self-optimizing, and self-configuration of the network. Similarly, in [96] users act selfishly to choose the wireless channel and power level to maximize their throughput, which in fact represents a distributed manner for the knowledge plane. However, the information can be used at the centralized controller to make optimal decisions when congestion in the network is high. Hence, both the distributed and hybrid architectures of the KDN are suitable for this study.

In [97], the authors integrated two DRL techniques for power control between primary users (PUs), SUs, and wireless sensors. Their work is based on an asynchronous variant of the actor–critic learning algorithm, where an A3C-based power allocation method and distributed proximal policy optimization (DPPO)-based power control are utilized. These two methods are based on the actor–critic learning mechanism for optimizing power control policies and making spectrum sharing more accurate. The aim is to model the information interaction between users and wireless sensors to learn the simultaneous power allocation scheme and optimize power consumption. In their algorithm, first, each SU gathers information from a centralized controller, and based on the experiences of other SUs, it performs power control strategies and simultaneous power control management policies at the controller. PUs cannot obtain power allocation policies using other PUs. They only adjust their transmission power based on a power-control scheme. Finally, the simulation results indicate that the proposed power control scheme performs better than the DQN-based power allocation in terms of power efficiency, spectrum sharing, and network convergence.

To allow spectrum sharing, the authors of [98] proposed an optimization-based algorithm for power management at small BSs (SBS) with a Q-learning method to reduce the interference of each RB. Initially, power is randomly allocated by SBS, based on the assumption that the SBS is concerned about maximizing its expected data rate in the long term, Q-learning is used to find the optimal power for SBS. The state of each SBS is a binary value that indicates whether the QoS has been violated, while the action of the learning algorithm is the selection of the optimal power level. The system reward is represented as the instantaneous rate of the SBS. Simulation results show the ability of Q-learning to increase the long-term expected data rate of SBSs. The centralized power control mechanism in [97] represents a centralized KDN, where the latest power control strategies are updated. The power allocation scheme in [98] relied only on local information at the SBSs. Local data are not shared within the SBSs. Hence, the algorithm can be applied to the distributed architecture of the KDN.

In addition, a research study in [99] utilized the RL technique to self-organize the transmission power in femto-BS, pico-BS, and micro-BS. Their goal is to mitigate interference in SCNs and increase the spectral efficiency. Here, the interaction between SBS and macro-BS is modeled as a non-cooperative game, where a joint utility and strategy estimation algorithm is proposed to help the SBS reach the desired equilibrium network operating point. In each iteration, the femto-BS selects an action based on its current strategy and receives a reward representing the data rate. The reward of the algorithm ensures that the QoS of the macro-BS users is satisfied. The self-organizing algorithm for interference management is fully decentralized, which suggests a distributed KDN architecture.

6.1.3. Quality-of-Service

Interest in the application of wireless networks is increasing daily. Hence, the future of wireless technology requires better and more reliable communication. Improving QoS with complex and colossal data traffic has always been a challenging research problem. As a result, having prior knowledge to enhance the QoS based on ML algorithms is crucial to ensure scalability. In this section, recent approaches for optimizing QoS based on supervised and reinforcement learning are explained. Table 5 summarizes the studies surveyed in this section.

(i) *Knowledge derived from supervised learning*: In [100], an SL-based QoS assurance architecture was introduced for 5G networks. Their algorithm uses past QoS information and learns the network environment and dynamic changes. The QoS prediction in this study uses a decision tree strategy to estimate the future QoS of the network. The proposed model consists of a layer called the history QoS data repository that collects all the information for training. The information collection comes from two parts: QoS data collection and QoS data preprocessing. Two methods are used for data collection: passive and active, enabling intelligent decision-making and quick QoS optimization across a wide range of applications. The rest of the system consists of modeling, training, QoS anomaly detection, and prediction. Based on the output of the anomaly detection and prediction, the system determines the QoS policies. Consequently, using trained SL to improve the QoS is beneficial, and the network can leverage the acquired knowledge produced by the ML to make appropriate decisions and predictions. Using an extract-transform-load (ETL) allows us to collect QoS data from heterogeneous networks and turn them into understandable and easy to comprehend structures. Therefore, it is vital to have extensive knowledge about network behavior and implement an algorithm to change based on the collected data. However, the drawback of this approach is the amount of data stored in the system repository. The algorithm is suitable for centralized KDN, where information regarding the QoS is available for automatic anomaly mitigation and action suggestions.

(ii) *Knowledge derived from reinforcement learning*: In [101], a universal deep neural model (DNM) for predicting multiple-attribute QoS is presented. Two loss functions are used to make accurate QoS predictions: the least absolute deviations and least square errors. During the training, gradient descent was used to determine the optimal learning parameters. The predicted QoS values are the knowledge extracted from this algorithm. After the training procedure, their method was compared to the mean absolute error (MAE) and root mean square error (RMSE) for prediction accuracy. The proposed technique achieves a higher prediction accuracy compared to the two methods. Their method can be used in a centralized KDN for a fast user's QoS prediction, where DNM is deployed in the KP to process information and generate useful knowledge in terms of QoS prediction.

Table 4
Knowledge-based strategies for power management.

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[89]	Real-time power allocation	Centralized	Supervised learning with DNN	Power control and interference management	Decreasing the computational cost but cannot be utilized for directional antennas
[92]	Adaptable to large networks with real-world scenarios	Distributed	Deep Q-learning	Suboptimal power allocation with faster convergence compared to WMMSE and FP	Robust to unpredictable changes of the wireless medium as well as delayed/incomplete CSI, although it lacks global knowledge that might degrade system performance
[95]	Smartly offload traffic and autonomous optimization of cell range expansion (CRE) in heterogeneous networks	Distributed	Reinforcement learning	Significant improvement in throughput	Joint power allocation and cell association by achieving target SINR for each UE but some backhaul constraints
[96]	Intelligent joint selection of power level and spectrum channel by D2D users in a multi-cell network	Distributed or hybrid	Multi-agent Q-learning	Reducing the power consumption and maximizing throughput	Near-optimal and fast convergence for D2D pairs, although it does not support multi-hop D2D
[97]	Simultaneous power allocation in wireless sensor networks	Centralized	Actor–critic based learning with DNN	Efficient power management with less frequent updates	Guarantees QoS but does not involve mobility
[98]	Trained Q function is capable of adjusting the SBS transmission power	Distributed	Distributed Q-learning	Maximize the data rate of SBSs while keeping acceptable QoS	Optimization of power strategies for maximizing the long term data rate
[99]	Self-organization and optimization of throughput by the femtocell user equipment (FUE) under QoS constraint	Distributed	Joint utility and strategy estimation based learning	Achieving NE with less overhead	High spectral efficiency when utility function is the network performance of FBSs, but the network is static during the learning process

Table 5
Knowledge-based strategies for QoS.

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[100]	Automatic QoS prediction for correction and suggestion for 5G networks	Centralized	Decision tree	reliable QoS	A new architecture for QoS prediction and reaction to dynamic changes of network
[101]	Efficient QoS prediction strategy	Centralized	deep neural network	High QoS prediction accuracy	Proposed algorithm has a quick response-time, which crucial for autonomous systems

6.1.4. Base station switching

One significant difference between 5G and 4G is the network architecture and deployment of a large number of SBSs. Because of mmWave signals in 5G cellular networks, BSs need to be closer to users to reduce the propagation loss and improve the channel capacity. However, deploying a large number of SBSs comes at a price and a significant increase in the total energy consumption of the wireless network. One of the promising solutions is BS on/off switching, which saves approximately 36 million kWh per year [102]. Considerable effort has been dedicated to finding the best strategy for on/off switching mechanisms in 5G wireless networks. Among them, ML algorithms have attracted attention for their self-optimization and self-management abilities. The extracted knowledge from a trained ML can be used in a centralized KDN architecture to manage the entire network BS on/off switching. Table 6 summarizes the studies surveyed in this section.

Knowledge derived from reinforcement learning: The authors of [103] proposed a Q-learning-based algorithm for heterogeneous networks to reduce the overall energy consumption of SBSs. Users' information is utilized with a heuristic algorithm for the implementation information case (HAIIC) with an offline solution to reduce energy consumption. In this work, the on/off switching of SBSs is discussed by gathering complete and incomplete information, which correspond to the future and current information, respectively. With complete information, the critical sections of a cell are defined. Based on an offline approach, the cumulative energy consumption of the SBS is obtained, and the best policy is attained. For incomplete information, the HAIIC was used to categorize the on/off switching policy of the BSs. The HAIIC uses an upper-bound threshold based on the energy consumption ratio (ECR). The learning algorithm defines the SBS on/off switching operation as an action. The state depends on the number of active users and SBS. Finally, the reward of the learning algorithm is achieved through a switching action in any particular state. The reward of the algorithm is entitled to energy consumption and transmission gain constraints. The procedure is iterative, and the calculated reward updates the Q-value until it converges. From the simulation results, it is concluded that the proposed HAIIC algorithm minimizes energy consumption. Similarly, studies were conducted, such as the work in [104], which used actor–critic learning to control the on/off BS switching. The RL technique defines the BS switching operation as the action and the amount of traffic load on the controller as the state. The controller decides an action based on the traffic load in a stochastic manner to minimize the overall energy usage. The authors of [103] used the coverage of the BS to switch off the SBSs in their vicinity, which suggests a centralized KDN architecture at the main controller. Similarly, in [104], the overlap between the coverage areas of BSs is considered to turn on/off a BS, which also suggests a centralized architecture of KDN.

Although the studies mentioned above used on/off procedures and significantly decreased power consumption, they lack the on/off state transition between BSs. To this end, the authors of [105] included the transition overhead in the cost function of the DRL-based framework for downlink scenarios in cloud radio access networks. The state in DRL consists of two components: the on/off state of the BS, which is a binary value (0 for off, 1 for on), and users' data rate demands. The action is taken based on the state of the iteration, which leads to the activation of a BS. The reward is calculated based on power consumption and user demand constraints. Simulation results indicate that the proposed advanced technique can be adapted to dynamic environments and can provide power consumption optimization while satisfying user demands. Moreover, the study in [106] considered the on/off transition state, where it uses the Q-learning method and a novel dual-threshold-based sleep-mode control for SBSs. The use of a dual threshold for controlling the BS sleep mode minimizes energy consumption and avoids frequent BS transitions. There is an upper user and a lower user threshold, which defines the action of the learning algorithm. The state of each SBS is the number of users under coverage. When the number of users in the cell passes the upper threshold, the small cell is switched on, and once the number of users becomes less than the lower threshold, the small cell is switched off. Based on the simulation, the algorithm achieves near-optimal performance by reducing the energy consumption with a minimum BS on/off transition state.

Moreover, a BS active/sleep scheduling scheme is proposed for k-tier heterogeneous networks, guaranteeing coverage, QoS, and throughput [107]. They used a fuzzy Q-learning method to put the BS in sleep, while there was no user to serve or activate a BS once the user was detected in the cell. To save energy, the algorithm uses an optimal sensing probability strategy for user detection. An SBS is in the sleep mode state when there is no active flow, and it randomly activates to scan the coverage area for possible users based on the tuned output sensing probability action. It is observed that the proposed algorithm can efficiently handle user population fluctuations and increase the energy efficiency. All three [105–107] proposed schemes can be used in a centralized architecture of KDN to provide an energy-efficient algorithm to reduce energy consumption.

6.1.5. Cache management

As predicted by Cisco [108], wireless networks, especially cellular networks, will produce about 30.6 exabytes of data traffic each month. This is due to the proliferation of smart devices and the appearance of high-tech applications, such as ubiquitous social networking, augmented reality, and high-definition live streaming. Faced with unprecedented data traffic, intelligent learning-based caching strategies have been introduced to alleviate backhaul traffic and shorten latency [109]. In this section, we investigate ML algorithms to assist in creating knowledge in the KDN paradigm for cache management. Table 7 summarizes the studies surveyed in this section.

(i) *Knowledge derived from supervised learning*: To construct estimation methods to identify the popularity of content in cellular networks, an ELM neural network is used to improve the QoE and reduce the network traffic [110]. The proposed method adapts mixed-integer linear programming for content replacement. The algorithm uses a perturbation stochastic approximation to select the physical cache size simultaneously and performs efficient cache deployment. In their method, the stochastic approximation reduces the number of neurons for ELM while ensuring an accurate prediction of future content popularity. The proposed method utilizes real-world data while making efficient cache decisions compared to the most popular cache deployment schemes, such as K-nearest neighbor [123] and regression [124]. The proposed caching scheme in [110] considers content popularity, cache size, network topology, and link capacity to perform efficient content caching and cache deployment. The algorithm has a cache manager to communicate with all BSs, making it suitable for centralized KDN.

The authors of [111] proposed a joint optimization of user association and content placement in an unmanned aerial vehicle (UAV) network. The goal of the optimization algorithm is to maximize the QoE of the user while minimizing the total transmit power of the UAVs. To predict the users' mobility patterns and content request distribution, ESN is utilized to learn the distribution from the provided training data (user traffic data). More specifically, the data are human-centric information, such as gender, users' visited locations, and requested content. The information is fed as a vector to the ESN input, where the output of the ESN obtains the content request probabilities. The conceptor-based ESN model enables C-RANs to separate user behaviors into independent patterns, which increases the prediction accuracy of cache management and power control systems. A similar study was conducted in [112] to predict the mobility pattern and content request distribution of each user in C-RANs. The objective is to minimize the delay and traffic in the network while caching the most suitable content. The algorithm first uses an ESN-based scheme to utilize the BBUs to learn the user's mobility pattern and content request. Then, a sublinear approach enables the BBUs to determine which contents to cache. The conducted simulation was compared with random caching with clustering and random caching without clustering, and it showed that the sum effective capacity reached up to 27.8% and 30.7%, respectively. In [111] the entire network is connected to the cloud's pool for information extraction, which represents knowledge for the entire network for centralized cache management. The approach in [112] for content request prediction and mobility pattern recognition is used via centralized BBUs and cloud heads, representing a novel approach for the centralized architecture of KDN to predict the content of mobile users in cellular networks.

The authors of [113] used three different methods to minimize the backhaul load by predicting the cache content from raw video data. First, a 3D CNN is used for feature extraction, in which a single frame of the video is analyzed alternatively. Second, the SVM algorithm is utilized for generating representation vectors of videos, and third, a regression model predicts the video popularity content. After obtaining the popularity of videos, the optimal segment of each video is cached by the BS to minimize the backhaul traffic. The proposed algorithm predicts the popularity of a new video without any statistical information. Here, the BS is connected to the core network, where the core network is connected centrally to the content server, which represents a centralized cache strategy.

(ii) *Knowledge derived from reinforcement learning*: To mitigate the challenges of content caching under spatio-temporal traffic demands, the authors of [114] introduced a decentralized caching

Table 6
Knowledge-based strategies for BS switching.

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[103]	Offline trained solution to minimize energy consumption in heterogeneous networks	Centralized	Q-learning	Reducing the cumulative energy consumption	Optimal small cell switch policy but startup energy cost and user mobility are not considered
[104]	The trained data is utilized at the controller to choose the active BS based on the traffic load in RANs	Centralized	Actor critic based learning	improving the energy efficiency	Robust BS switching solution but eliminated inter-cell interference
[105]	Dynamic resource allocation in cloud RANs	Centralized	Deep reinforcement learning	Minimizing power consumption	Adaptable to dynamic environments while satisfying user demands
[106]	Using the trained data for energy-efficient and QoS-aware in SBSs	Centralized	Q-learning	Minimizing network energy consumption while avoiding frequent mode transition in BS switching	Effective multi-objective energy optimization between BS switching and QoS, no user mobility is considered
[107]	Self-organizing the active or sleep mode of SBS in heterogeneous networks	Centralized	fuzzy Q-learning	Improving energy consumption while maintaining network capacity and coverage area	A thorough BS switching based on user activity fluctuations, QoS, channel propagation, and interference

update scheme among different small cells based on joint utility and strategy estimation. The SBSs can optimize the time-varying caching probability distribution using the received instantaneous utility update. The proposed algorithm is capable of minimizing the service delay when serving user requests at the SBS. Similarly, Wei et al. focused on distributed caching design at BSs to reduce the traffic load via D2D communications [115]. BSs can cooperate and exchange information regarding their locally missing content from other BSs through the backhaul channel to make the scheme more cost-efficient. A Q-learning algorithm is utilized to improve the system transmission cost and enable D2D devices to offload traffic using cache utilization. The action of Q-learning depends on the adjustments of the cache contents taken in a specific state for observation. The convergence of the proposed cache replacement strategy is tested by the sequential stage game model, meaning that the decision-making process of each BS on cache placement is a cooperative game (every state can be considered as a different stage). The simulation results have shown superior performance compared to conventional strategies, including the least recently used (LRU) strategy [125], least frequently used (LFU) strategy [126] and randomized replacement strategy [127]. In [114], each SBS optimizes the cache policy via a decentralized cache strategy, which represents a distributed architecture of the KP for future optimization via KDN. In [115] the traffic is offloaded from the cellular channel to the WiFi channel using D2D communication between users, which provides a distributed architecture for users in the network to exchange data traffic.

Cache-enabled D2D communication technology is expected to lower the requested content and congestion at the BS by enabling devices to request content from nearby users. The authors of [116] focused on joint content delivery policy and cache content placement. Cache content placement determines the amount of traffic unloaded from the BS to the D2D. This study uses RNN methods, specifically ESN and long short-term memory (LSTM), to predict the users' mobility patterns and content popularity. Therefore, the algorithm realizes where to cache and which content to cache. Once the user's local cache content cannot satisfy the content request of the user, the user will establish a D2D link with one of the neighboring users. The process of selecting the most appropriate user was performed using a DRL agent. The DRL agent optimizes content delivery and makes dynamic decision making for user selection. The simulation results indicate that the proposed content placement and content delivery approaches improve the cache hit ratio and reduce delivery delay and energy consumption. This method is inherently distributed, where nodes find their requesting content from nearby users. However, this information can be used in a centralized manner to monitor the content exchanged and minimize the network traffic on the BS.

Owing to the time-varying nature of the wireless channel, the authors of [117] proposed a deep reinforcement approach to optimize the cache-enabled interference alignment. They adapted a finite-state Markov channel (FSMC) and used a deep Q-network to develop a caching update scheme. First, a centralized scheduler collects CSI and cache status information from the users. Then, a deep Q-network feeds the optimal action for the current instant and stores the agent's experience of each time instance to optimize the Q-network parameter accordingly. Their method maximizes the Q-learning reward function based on existing knowledge and attempts new actions to acquire new knowledge. The proposed algorithm significantly improves the network performance compared to other studies, in which an invariant channel is assumed. This is a promising technique to be used at KP in a centralized manner, where a controller collects the CSI from the users and sends information to the deep Q-network to obtain the optimal policy for users. The output of the learning algorithm is the knowledge information used in response to wireless networks with similar behavior.

The authors of [118] utilized the DRL algorithm with the Wolpertinger [128] architecture for content caching at the BS. The objective of the algorithm is to maximize the long-term cache hit rate with no information about the content popularity distribution. The input of the DRL system is the requested frequency of files and the current file request, and the action of the system is whether to cache the requested content at the BS. The proposed technique is compared with different traditional cache update schemes, namely LRU and LFU [129]. The simulation results prove that the algorithm outperforms both schemes in terms of short-term and long-term cache hit rates. In [119], the authors proposed a DRL approach to enable dynamic orchestration of caching resources, networking, and computing

Table 7
Knowledge-based strategies for cache management.

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[110]	Adaptive caching scheme for cellular networks	Centralized	Extreme learning machine	Reduces the network traffic and increases the users' QoE	The proposed scheme outperforms industry standard caching schemes
[111]	Optimal cache-enabled method for UAVs	Centralized	Echo state networks	Maximizes users' QoE and minimizes the total transmission power	The proposed algorithm locates the UAVs and predicts the content to cache at UAV
[112]	Proactive cache approach in cloud-based radio access networks	Centralized	Echo state networks	Maximizes the sum effective capacity	The proposed method also decreases the delay and traffic in the network
[113]	Applying intelligence-based content-aware in cellular networks	Centralized	3D CNN, support vector machine and regression model	Improves the backhaul load	The proposed scheme is content-aware and cost-effective
[114]	Optimization and continuous update of caching policies in decentralized manner in small cell networks	Distributed	Joint utility and strategy estimation	Minimizes the service delay	The proposed scheme shows 15% and 40% gain compared to other baselines
[115]	Dynamic improvement of cache efficiency in cellular networks	Distributed	Q-learning	Minimizes the transmission cost	The algorithm is capable of D2D offloading and cache replacement
[116]	Intelligent joint content placement and content delivery using D2D communication	Distributed or hybrid	Deep reinforcement learning and recurrent neural network	Improves the cache hit ratio	The proposed strategies and approaches can efficiently reduce the energy consumption and delivery delay
[117]	Intelligent and realistic cache-enabled and interference alignment for next generation wireless networks	Centralized	Deep reinforcement learning	Improving energy consumption and the network's sum rate	Saving the limited backhaul capacity using cache-enabled opportunistic interference alignment
[118]	Content catching with a single BS scenarios	Centralized	Deep reinforcement learning	Maximizes both long-term and short-term cache hit rate	Reducing the computational complexity, but it only incorporates one BS
[119]	Intelligent joint optimization of networking, computing and caching resources in the next generation vehicular networks	Centralized	Deep reinforcement learning	Improves the traffic control and network efficiency	Joint optimization of resources and cache content
[120]	Knowledge transfer from D2D interactions to improve content popularity matrix in the target domain	Centralized	Transfer learning	Increasing the users' QoE and improving backhaul capacity	Optimal cache strategy at small cells for estimating content popularity, traffic load and backhaul capacity
[121]	Sharing prior knowledge from D2D communication	Centralized	Transfer learning	Maximizes cache-hit ratio	Improves the prediction task and caching performance
[122]	Efficient caching mechanism for heterogeneous networks	Centralized	Transfer learning	Reduces the training time	The proposed scheme can efficiently estimate the popularity profile

resources in vehicular networks. The DRL agent assigns vehicles to BSs and decides whether to cache the requested content by the vehicle at the BS. The proposed algorithm jointly optimizes the problems associated with resource allocation and caching. Simulation results with various system parameters show that the DRL system performs much better than existing methods, such as edge caching, mobile edge computing (MEC) offloading, and virtualization. Based on the requested content in [118] the DRL agent acts as a controller to decide whether to store the content at the BS local storage. This method suits the centralized architecture of the KDN to serve users directly with a minimum delay. A more sophisticated strategy was adopted in [119] to perform caching, computing, and networking in a systematic, centralized manner using the SDN controller, which is a close research study to initiate knowledge to accomplish different networking tasks.

(iii) *Knowledge derived from transfer learning*: In [120], contextual information from D2D interaction, including users' social ties and content viewing history, is observed to execute a TL at each SBS in the network. Their method improves backhaul offloading based on the popularity of the content. This information comes from source domain sharing and accessing D2D data between users in a social community. The information is later used as prior knowledge for a content popularity matrix for estimation in the target domain. Moreover, they utilize the traffic load and backhaul capacity as feedback to the system for further improvements. From the simulation results, it is clear that the proposed TL cache procedure enables wireless systems to have prior knowledge instead of starting from scratch and also deals with data sparsity. Generally, TL acts as a knowledge plane and provides essential information to networks with similar behavior. Using caching optimization using TL is a significant step toward KDN, and one of the significant breakthroughs in this method is the maximization of QoE. Similar work was conducted in [121] to extract knowledge from the interaction between users by accessing, sharing, and recommending files. Instead of learning from scratch, the information obtained via D2D communication among users is used for caching content at the network edge. The proposed TL approach increases the cache-hit ratio and outperforms classical collaborative filtering (CF) methods [130]. Both studies provide centralized knowledge for the network to reduce backhaul traffic and improve content management.

In heterogeneous networks, SBSs are assumed to have high storage capabilities to cache popular files, such that the user can capture the file in a faster and more efficient manner. In particular, SBSs must distinguish the popularity of files and estimate the user demand for that file within a specific time interval. The authors of [122] proposed a TL-based approach to increase the performance of estimating a popularity profile. The centralized approach uses prior knowledge to compute and estimate the popularity of a file based on requests during a predefined observation period. The estimation was then used to optimize the catching probability. The proposed approach reduced the convergence time of the training phase. In this study, centralized knowledge is generated based on the popularity profile of the cache content.

6.1.6. Backhaul management

Content caching at SBSs requires backhaul management, and because of the heterogeneity of backhaul, both wired and wireless backhaul must work together to handle the massive traffic. Wired links use fiber cables, and wireless connections are now deploying mmWave frequencies. Owing to the heterogeneity of backhaul links, the management of backhaul has attracted significant attention. Different solutions have been proposed to reduce the complexity of backhaul [131,132]. However, new studies have concentrated on ML for reliable backhaul management. In the following section, ML studies that can be deployed in the KDN are investigated. Table 8 summarizes the studies surveyed in this section.

Knowledge derived from reinforcement learning: The deployment of low-power and short-range heterogeneous SBSs and the exponential increase in wireless traffic will cause congestion in backhaul for BSs to communicate with each other. The authors of [133] designed a distributed backhaul management model from a game-theoretic perspective. They utilized the RL technique to solve the gaming problem using joint utility and strategy estimation. Each SBS is responsible for predicting files to download without compromising the required transmission rate. In this study, different SCNs with several coexisting backhaul systems are connected using wired links, mmWaves, and sub-6 GHz frequency bands that can only support a few files per time slot. This work delivers a balance between downloading files and provides an equal chance for SBSs to access backhaul. The self-organizing nature of the RL algorithm allows SBSs to reach a Boltzmann–Gibbs equilibrium that approximates the desired proper mixed Nash equilibrium (PMNE). In [134], the authors proposed adaptive cell range extension (CRE) and backhaul management based on RL techniques to set the CRE offset values autonomously. The CRE offset value affects both cell selection and handover, where this information is available to all concerned cells on the access and backhaul interface. In the proposed algorithm, the state value of each small cell is obtained via backhaul capacity and whether it is violated, and the action is the cell expansion or CRE bias of a cell considering whether backhaul capacity is available or not. To maximize the total backhaul capacity, a cost function is defined for each small cell with different constraints to satisfy the capacity. Moreover, the cost function is optimized using Q-learning via an iterative approach. Simulation results demonstrate the improvement of network backhaul throughput in macrocells and the QoE of users. Xu et al. [135] proposed a distributed load balancing scheme based on a Q-learning algorithm to improve the backhaul resource utilization in heterogeneous networks. The backhaul utilization is quantified to different levels that represent the state of the system, and each BS determines the bias value as an action. Then, based on the acquired reward obtained from the weight difference between the outage probability and the backhaul capacity, the bias value is updated. Therefore, Q-learning optimizes the bias value selection strategy to balance system-centric and user-centric performance. The results from the numerical analysis show that this algorithm is capable of efficient utilization of backhaul resources, which guarantees load balancing and improvement in QoS for each user. Solving backhaul constraints in future wireless networks is an important factor for cache management and load balancing. The majority of the research works, and the three backhaul management studies investigated here [133–135] can serve as centralized knowledge for KDN architecture.

6.2. Networking

With the exponential growth of users and data traffic in the network, networking in wireless communication systems requires more advanced solutions. In particular, challenges including the imbalanced distribution of traffic loads among BSs and wireless channel dynamics need to be addressed. Furthermore, emerging vehicular networks and self-driving vehicles introduce new difficulties that are not addressed in traditional networking algorithms. To overcome these issues, the knowledge acquired by ML algorithms can assist networks in building intelligence and automation. Therefore, new ML studies for routing, clustering, data aggregation, and user associations were investigated.

6.2.1. Routing

User demands for high-resolution data with enormous sizes have led to new RAN technologies, including cognitive radio networks, C-RANs, and ultra-dense networks (UDNs). To facilitate effective networking, routing strategies and policies play an important role. Traditionally, network upgrades rely on hardware solutions, for example, by improving the core size or increasing the router size to enhance network performance. On the other hand, the software development aspect of traffic management for routing policies has consistently failed because of the varying network environments. Recently, ML has made a major technological breakthrough with efficient routing protocols to enhance processing packets and throughput performance [136]. Accordingly, we provide a detailed summary of the novel ML techniques-based routing for improving the knowledge layer of the KDN to create self-configuration and self-optimization. Table 9 summarizes the studies surveyed in this section.

(i) *Knowledge derived from supervised learning*: In [137], an SL algorithm based on a meta-layer was proposed to solve dynamic real-time routing problems. In this study, a centralized SDN controller collects node information via the OF protocol. This information is fed to multiple ML modules in the meta-layer,

Table 8
Knowledge-based strategies for backhual management.

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[133]	Enabling SBSs to autonomously manage and optimize the traffic in 5G networks	Centralized	Joint utility and strategy estimation based learning	Minimizing the traffic congestion in the backhaul	The proposed solution converges to proper mixed Nash equilibrium (PMNE) (an approximate version of NE)
[134]	Running self-optimization mechanism on small cells for backhaul-aware cell range extension	Centralized	Q-learning	Backhaul management and optimized cell range extension	The proposed approach improves QoE and mitigates the backhaul traffic conflict in macrocell
[135]	Intelligent load balancing using backhaul resources	Centralized	Q-learning	Minimizing the probability of the user's outage and increasing the backhaul resource utilization	The proposed scheme improves the average throughput

which consist of heuristic algorithms. After the training process, the algorithm can identify the path by satisfying system QoS. Once the controller receives new routing requests, the trained ML instantly provides heuristic-like results. The system performance is compared to the classical max–min ant system (MMAS) [154] which has been proven to be an appropriate approach for examining the performance of routing frameworks. MMAS is an upgraded version of the ant colony optimization method taken from ant routing [155]. The proposed study [137] can be adapted to the centralized architecture of the KDN for this particular study. The reason is that the routing framework utilizes the SDN functionality to gather global information and then predict a route. Therefore, the proposed protocol is inherently centralized and suitable for the centralized architecture of O-RAN networks. The proposed technique enables the knowledge plan in the KDN to identify routes that maximize the network QoS.

An autonomous vehicle or self-driving car can communicate with other vehicles, roadside units, and infrastructure. This capability is known as vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication to exchange essential information, such as speed, location, environmental conditions, etc., to nearby vehicles and the controller. Authors of [138] proposed a delay-bounded routing framework for vehicular ad hoc networks (VANETs). They focused on delivering messages with user-defined delay parameters and minimum usage of the radio spectrum. The delay-bounded routing protocol uses linear regression to predict the traveling distance and available time for forwarding a message. Their algorithm has two schemes, the greedy and centralized schemes, which are both based on linear regression. The greedy strategy predicts the available time by using current sampling data, and the centralized scheme uses global statistical information to choose the optimal path for routing. The simulation results illustrate that the radio usage is greatly reduced. Moreover, the functionality of using both greedy and centralized-based techniques establishes a connection for the hybrid architecture of KDN to enable the installation of routing protocols in VANETs.

In [139], the authors combined two SL classifiers; decision tree learner and rule learner, for routing optimization in a wireless sensor network. They proposed a MetricMap based on MintRoute, which collects the routing protocol to obtain the link quality. MetricMap uses two components, the first component updates the features for the learning strategy when a packet arrives. The second component controls the link classification with input from the features, and the output values indicate the link quality. For their performance measurements, they considered data latency, data delivery rate, and fairness index. From the evaluation of the 30 sensor nodes in the network, the MetricMap achieves up to 300% improvement in data delivery with no effect on other performance matrices. As indicated in the study, the training phase is made at a backend server, which suggests a centralized unit that collects all the information, trains a machine, and then instructs the nodes. This means that a centralized knowledge plane is created in this study to guide the nodes. The reason for a centralized KDN is that the authors introduce a two-stage route prediction where the first stage updates the learning strategy, and the second stage identifies the route. The learning strategy requires global information to converge to optimum performance. Therefore, a centralized KDN architecture can provide better information than a distributed architecture. Moreover, the first stage of the learning algorithm is an excellent example of deploying a machine learning in the knowledge plane. The controller uses the second stage to prescribe routing decisions for the network layer.

Sharma et al. [140] proposed a routing protocol for efficient routing in opportunistic networks called MLProph, which uses a decision tree and NN algorithms to compute the node's successful delivery probability. The algorithm uses past network routing data to compute the probability of whether the data will be delivered to the destination by the relay nodes. The probability value helps to decide on the next-hop selection. The ML method trains itself based on different network characteristics, such as node density, buffer capacity, hop count, node energy, mobility rate, and number of successful deliveries. Here, the NN is trained to determine whether the forwarded message has been delivered; it can have two outputs p=1 for successful deliveries and p=0 otherwise. The NN is trained iteratively by setting initial values, and subsequently, it provides optimal predictions for successful and unsuccessful deliveries. The simulation results indicate that the proposed algorithm performs much better than previous works in terms of overhead ratio, average latency, and packet delivery ratio. They also compared the two ML models and found that the NN performs better than the decision tree in terms of overhead, delivery probabilities, and packet loss. As much as the algorithm looks more distributed, it needs to be fitted to a centralized KDN owing to the next-hop selection strategy. The proposed protocol uses the successful data delivery of the nodes to make routing decisions. In distributed frameworks, nodes might not be able to monitor the whole network and build routing tables because the information they can collect is limited. Hence, the proposed protocol is more suitable to be used in a centralized framework. Moreover, this strategy depends on various attributes, such as the distance from the source node to the destination, current hop count, mobility rate, and energy, all of which are available at a centralized controller.

To improve the traditional routing strategies and increase the performance of the wireless backbone, the authors of [141]

Table 9
Knowledge-based strategies for routing.

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[137]	Real-time route discovery by the trained dataset	Centralized	Deep neural network	Improves network performance by efficient traffic engineering	Simple framework and computationally efficient, but it lacks some network features including node mobility, backhaul traffic, etc.
[138]	Routing strategy can be used for similar VANETs	Hybrid	Linear regression	Efficient spectrum usage and high packet delivery ratio	Introducing delay-bound routing protocol with accurate route prediction with high efficiency
[139]	Optimized route selection in wireless sensor networks	Centralized	Decision tree and rule learner	Significant improvement in data delivery ratio	Efficient sensor communication via user link quality
[140]	Enabling efficient route selection	Centralized	Decision tree and neural networks	Decreasing average latency, improving overhead and packet delivery ratio	Highly efficient route selection based on various network parameters
[141]	Real-time updating and routing judgments in heterogeneous networks	Centralized	Supervised learning and deep CNN	Minimize the average delay and improves packet loss ratio	Intelligent traffic control
[142]	Self-driving networks by learning traffic control mechanism in heterogeneous networks	Distributed	Supervised deep neural networks	Better signaling overhead, delay, and throughput compared to OSPF	Capable of learning complex patterns and functions to predict the least cost path
[143]	Efficient and intelligent route decisions in wireless mobile networks based on KDN architecture	Centralized	Principle component analysis and neural network	Lower packet loss ratio and acceptable throughput and E2E delay	Provides a dimension-reduction vector matrix to reduce the algorithm response time but it must be verified over larger networks
[144]	Trained algorithm can select next hop with a least total average hop count and successful delivery probability in wireless networks	Centralized	K-mean clustering	Less dropped packets and network overhead	Simple, efficient routing protocol but needs improvement in average message latency and should involve energy consumption as a node feature
[145]	Optimal route selection and prediction capability using global information for VANETs	Centralized	Unsupervised learning-based algorithm	Better transmission delay compared to existing VANET routing protocols	The proposed scheme is robust to varying mobility rates
[146]	Smart routing decision in IoT-based smart cities	Centralized	Deep reinforcement learning	Mitigating the network congestion and load balancing	Simultaneous QoE satisfaction and crowd management
[147]	Finalized updated route table can be used as the routing policy in wireless sensor networks	Distributed	Q-learning	Extending the network lifetime	Acceptable communication and computational overhead, it can be extended with more node characteristics, such as mobility and traffic
[148]	Identifying stable routes in cognitive radio networks	Distributed	Q-learning	Minimizing the interference between SUs and PUs and less frequent route discovery	Boosting network scalability and functionality
[149]	Route selection for multi-hop cognitive radio networks	Hybrid	Reinforcement learning	Improves the QoS	Selecting the best possible route in terms of throughput and packet delivery ratio
[150]	Efficient route selection in dense cognitive radio ad hoc networks	Distributed	Reinforcement learning	Minimizing the interference	Stable protocol when the network size increases but might be degraded in high mobility scenarios
[151]	Intelligent QoS-aware route selection	Distributed	Q-learning	Energy efficient, QoS-aware and mobility tolerance	Reliable, stable and extended lifetime network

(*continued on next page*)

Table 9 (*continued*).

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[152]	Intelligent traffic routing control with KDN approach for next generation of wireless networks	Centralized and hybrid	Deep reinforcement learning	Minimizing the overhead	The proposed paradigm combines distributed and centralized intelligence to achieve highest performance
[153]	Automatic adaptation based on the traffic conditions via KDN paradigm	Centralized, distributed, or hybrid	Deep reinforcement leaning	Traffic engineering	Near optimal solution after one single training step

proposed an intelligent routing scheme using a deep CNN. Their method learns from the previous experience based on congestion, and uses this information to train a two-phase procedure, namely cold start period and intelligent running period. The cold start period is the initialization of the training set, where the algorithm only defines a route with a minimum hop path. After this period, the algorithm switches to the intelligent running period, where it performs real-time updating and routing judgments. More importantly, a CNN is constructed for each routing decision, which takes the collected information based on traffic patterns from routers, including traffic generation rate, to predict whether the selected routing strategy can cause congestion in the network. This process is periodically updated until it is predicted that the chosen route will cause no congestion. Simulation results prove that the proposed algorithm performs much better in terms of E2E delay and packet loss ratio compared to conventional routing strategies, where there is no intelligence. The proposed method is a real-time intelligent network traffic control method that can be adopted in a centralized KDN. A centralized KDN can collect traffic congestion information from the network and tune the deep CNN to converge to optimum performance within the timeframe threshold. It is not recommended to use the distributed knowledge plane due to the algorithm's real-time updates, which require the maximum amount of data from a large number of nodes to make appropriate routing decisions.

In [142], a supervised DNN was proposed for routing optimization in heterogeneous networks to predict the path from the source to the destination node. Each router in the network uses a DNN to predict the next hop; the DNN takes traffic patterns as inputs, and based on these inputs, it generates the desired output. The output of the deep learning structure significantly improved the network traffic management. There are three phases to obtain a fully functional ML. The first phase is the initial phase, where the traditional routing protocols, such as OSPF, provide the network route, and the network starts to operate. At the same time, the second phase is the training phase to train the deep learning system from the collected information based on the traditional operating system. Finally, the running phase is the stage in which the machine is fully trained and can provide real-time routing strategies. This method has been proven to have a higher throughput and lower overhead compared to OSPF. The proposed study suggests a greedy-based distributed architecture over a knowledge-based network to increase the throughput.

(ii) *Knowledge derived from unsupervised learning*: In [143], the authors focused on load balance routing based on PCA and NN for dimension reduction and prediction of the network load status. To obtain intelligence from the network, they combined SDN with ML and data analytics. The use of these algorithms has led to efficient and intelligent routing decisions. This article aims to address the shortcomings associated with the next generation of wireless mobile networks, such as video streaming and online gaming, to mitigate the delay caused by traffic. They proposed a routing strategy based on an ML scheme, where PCA was used to reduce the dimension of the vector matrix by applying it to the original adjacency matrix of the network topology. Based on the normalization, they designed a queue-utilization routing algorithm for routing prediction. Moreover, routers were continuously updated based on neighbors' information to select the routers with more resources. In this vein, they explored the current SDN architecture and represented an ML algorithm to predict routes. Nodes can also use the proposed algorithm to reduce the unnecessary information in the routing table and decrease the contention on the nodes in a distributed manner. However, this framework represents a centralized KDN, where the controller collects the information from the data plane and then uses ML techniques to obtain knowledge to identify an efficient route. Overall, the representation of SDN with knowledge suggests a centralized KDN.

Owing to the fixed network architecture of some routing protocols and the massive volume of data traffic exchanged between devices, the authors of [144] introduced a context-aware routing protocol named KROp. This protocol uses several network features to make routing decisions based on the network conditions. KROp uses the K-mean clustering algorithm and exploits network features to select the best next hop. This algorithm is based on the knowledge acquired from the node's behavior to identify a cluster of the best forwarders. The numerical results show superior performance in KROp in terms of dropped packets, overhead, and average hop count compared to other routing strategies, such as history-based prediction routing (HBPR) [156], and probabilistic routing protocol using the history of encounters and transitivity (PRoPHET) [157]. The proposed protocol uses the entire network information to make routing decisions. This means a centralized unit has the advantage of collecting more useful information compared to distributed. Therefore, this technique is suitable for a centralized KDN to make the next-hop selection decision.

Tang et al. [145] proposed a centralized routing scheme with mobility prediction (CRS-MP) for VANETs. Their method utilizes an SDN controller with an ANN to gather information and predict the user's arrival rate. Based on the arrival rate of each vehicle, RSUs or BSs can model statistical traffic patterns and estimate traffic mobility. Intelligence was also used in this study by integrating the CRS-MP model at the RSU/BS to predict the mobility patterns of vehicles and find vehicle connections. The ANN takes an input according to the number of arrival vehicles at different time instances, and based on the initial random weights, it predicts the vehicle arrival rates. The arrival rate results in the arrival rate function, which is later used to make routing decisions and evaluate the transmission rate and average delay. The numerical results of the CRS-MP scheme outperform other vehicular routing protocols, such as V2I and V2V communication, in terms of overall vehicular service delay. Furthermore, the proposed algorithm is independent of the mobility rate, making it more robust to high mobility rates. The proposed routing protocol utilizes multi-hop routing in vehicular systems using an SDN controller that solves the overload on the

BS. Hence, in the near future, centralized routing protocols for mobile devices can use the knowledge plane to decide whether the BS or other devices must route the packet. Centralized VANET protocols have better performance compared to the decentralized topologies. In centralized VANET protocols, vehicles are supported by the BS and road-side units (RSUs). On the other hand in purely distributed systems, vehicles must relay any vital information across the network using multi-hop communication. Therefore, there is a risk of link failure in highly mobile environments, which might have catastrophic consequences. Generally, VANET protocols are supported by a centralized controller to collect data and inform vehicles. The proposed protocol in [145] uses a centralized controller to collect data and process the data through ML algorithms. Consequently, centralized KDN architecture is usually more suitable for VANET protocols.

(iii) *Knowledge derived from reinforcement learning:* With the massive growth of IoT devices connected to the edge network, the design of routing strategies is complicated. In particular, in smart cities, routing is significantly more difficult owing to the distribution of the crowd and network congestion. The authors of [146] designed a DRL algorithm for smart routing decisions for load balancing and mitigating network congestion when massive crowds are moving around the city for daily activities. They adapted a DRL agent to directly use the NN and generate Q-values. First, the network state information is collected by the SDN controller on top of the network. Then, the DRL agent makes an action (routing decisions) based on the current state, and finally, the agent receives a reward. The objective of the reward function is to maximize the successful service access rate, minimize the data transmission delay, and balance the network load. The algorithm performance was better than that of the OSPF and enhanced-OSPF (EOSPF). Fig. 1 in this article represents the network architecture of the proposed algorithm, which shows a solid connection to the KDN centralized architecture, where all the information is collected at the controller and is processed by ML in KP. The knowledge/reward is later used as an action for the following routing strategy.

In [147], the study tackles the energy-aware routing in wireless sensor networks (WSNs) to transmit data packets using efficient paths within the shortest time such that the lifetime of the network increases. Specifically, they used Q-routing algorithms and extended them to propagate information faster with lower energy consumption [158]. The algorithm uses Q-learning to save the energy levels of the nodes in matrices after a sensor sends a feedback message. When the sensor receives the feedback messages from neighboring nodes, it modifies the Q-values in the achieved routing table. Once a node has a packet to transmit, it selects the next node from the routing table with the best Q-value in a greedy manner to relay the packet to the destination. Their technique has proven optimal routing decisions for low-energy nodes. The greedy-based approach allows nodes to individually select the best route, which suggests a distributed KDN.

Recently, CRN has attracted considerable attention owing to its importance in future wireless communication systems. This technology overcomes the scarcity of the channel spectrum by allowing secondary users or unlicensed users to benefit from underutilized licensed channels. However, the dynamic nature of CRNs makes routing a complicated task. The authors of [148] proposed a clustering mechanism or cluster-based routing to boost network scalability and functionality. Once the cluster heads are identified in the network, each cluster head estimates the Q-value of each neighboring node. The routing table is constructed based on the Q-values, and the largest Q-value is the next chosen node for the next hop. During the learning procedure, the state of the network represents the destination node, and the decision to select the next hop is the action. Finally, the throughput resulting from the chosen hop is the reward of the system. In this study, the knowledge is derived from each state and action pair, which provides an appropriate action for the next instant. The proposed cluster-based routing is recognized as a distributed method for routing a packet in a CRN. Because the Q-values are estimated locally by the nodes, the highest values are used to build the routing table. This means the routing mechanism is constructed in a distributed manner and is more suitable for distributed KDN architecture.

In [149], the authors studied three route selection schemes in a real testbed environment to improve the performance of multi-CR networks. One of the schemes is based on spectrum leasing, and the other two are based on RL. Spectrum leasing is a new term used for communication between unlicensed and licensed users in CR networks. The two RL algorithms are based on Q-learning to predict the next-hop neighbor. Similar to other studies [148] the next hop is selected based on the highest Q-value. The state action is the destination node and the selected next-hop node for the source node to transmit the data. The reward is the channel-state information. The proposed routing scheme was compared with the highest-channel (HC) protocol in a multihop network and has shown better performance. In the proposed study, we can adapt a hybrid KDN because both centralized and distributed models are utilized. The proposed protocol uses the channel data information to select the next best hop based on the output of the Q-learning. The channel information can be accessed by both the user and the BS, which means the protocol

In [150], the RL technique was applied to CR networks to obtain the influences of various network characteristics on the network functionality, while a routing scheme called cognitive radio Q-routing (CRQ-routing) runs in the network. CRQ-routing is a popular approach for minimizing interference between unlicensed and licensed users. The algorithm investigated the effects of Q-learning-based routing strategies on the network characteristics, including the network size. The outcome of their research shows that network size has a slight impact on the packet loss rate and end-to-end latency of SUs. Additionally, CRQ-routing proved that it has better network performance than other routing algorithms when the network size increases. Finally, they concluded that under different situations, such as changing node position and different routing lengths, Q-learning requires additional time to learn what degrades the network functionality. Concurrently, to manage the network overheads in highly mobility scenarios, the authors of [151] proposed the mobility, residual energy, and link quality-aware multipath (MR-LAM) for routing decisions. To do this, they used a Q-learning algorithm to select the optimal route with energy-efficient nodes. The proposed routing scheme aims to maintain a reliable and stable network during a particular timeframe. They have successfully improved several metrics including energy cost, end-to-end latency, convergence time, and packet loss ratio, when compared with other routing techniques, including the multipath optimized link state routing (MP-OLSR) protocol [159] and the extension version of MP-OLSR known as MP-OLSRv2 [160]. Both the proposed protocols in [150,151] act as a distributed network, where each node decides on the next hop. Therefore, both studies are suitable for the distributed architecture of the KDN.

The authors of [152] added intelligence to the network to mitigate the complexity of network topologies. They integrated both centralized and distributed network functionality to guarantee high QoS. Their hybrid approach uses AI routers for distributed intelligence and a network mind for centralized intelligence. AI routers are responsible for hop-by-hop IP routing to ease network overhead. The network mind acts as a global controller for connection-oriented tunneling-based routing protocols, including segment routing and multiprotocol labeled switching. A DRL-based routing strategy is deployed in the network mind for tunneling-based routing to ensure QoS. The state of the RL algorithm is represented by the network traffic characteristic information and device information, and the action is the forwarding path. The reward it acquires is the effectiveness of an action with respect to the delay requirements and optimization targets. Their proposed RL method converges to the global minimum, and the routing strategy shows better performance compared to other routing strategies in congested areas. The proposed scheme has a centralized architecture with an intelligent control plane. In our proposed framework, this plane controls and installs the rules in the network. Stampa et al. proposed a DRL algorithm for optimizing routing in a centralized knowledge plane [153]. The actor–critic learning method is used, where the state of the learning algorithm is calculated by the traffic matrix (which is defined by the bandwidth request between pairs of source and destination), and the action is the path taken to transmit data (obtained using link weights). Finally, the reward of the algorithm is based on the average network delay. Their method provides operational advantages compared to traditional optimization algorithms for routing strategies. This algorithm can be used in both distributed and centralized KDN architectures, suggesting a hybrid structure.

6.2.2. Clustering

In wireless networks, nodes/sensors/users have always been clustered to describe their distinctive features or differentiate based on their mobility rate, coordinates, etc. Clustering different nodes for different purposes improves the overall performance of the network. It is evident from the introduction of ML techniques that clustering problems are naturally solved using K-mean algorithms. However, other clustering methods have been proposed within supervised and unsupervised learning techniques. Clustering is one of the primary and essential applications of KDN for various purposes, such as traffic classification and data storage. Table 10 summarizes the studies surveyed in this section.

(i) *Knowledge derived from supervised learning:* One of the problems in ML is class imbalance, where the class distributions are highly separated. This means that the total number of minority or scares classes (also known as positive ones) is far less than the majority class (represented as negative) for a two-class scenario. When we apply a traditional classifier in these scenarios, they are likely to predict everything as a majority or negative class. In [161], the authors used logistic regression for imbalanced problems to improve the performance of the learning procedure. The proposed method is called logistic regression for imbalanced learning based on clustering (LRILC). First, K-mean clustering was applied to the dataset to partition the majority class into small clusters. Logistic regression is then used to overcome the class-imbalance problem. The experimental results show a higher accuracy in clustering the dataset compared to state-of-the-art classification methods. The proposed method can be used in a centralized KDN to solve imbalanced problems with large datasets.

In [162], fraud calls are identified by investigating the user's behavior. Their method uses the application of SVM alongside fuzzy clustering to identify fraudulent phone subscribers. Fuzzy clustering takes unlabeled input data and clusters the data according to their similarities. Furthermore, after a trained data algorithm obtains input data, it generates a value between 0 and 1. If the output value is closer to 1, it shows a higher degree of similarity. Their algorithm takes large datasets and utilizes PCA to reduce the dimension, and then uses the library of support vector machine (LIBSVM) and least square support vector machine (LS-SVM) with fuzzy c-means (FCM) and fuzzy K-means (FKM) to build the user's profile. If a call pattern does not match any standard pattern, it is classified as a fraudulent call. A comparative study was conducted using different methods of SVM and fuzzy clustering, as specified above. This shows that using LIBSVM and LS-SVM leads to better approximation and accuracy. The proposed algorithm can be utilized in a centralized BS to differentiate between ordinary and fraudulent calls. Accordingly, knowledge can be derived from this study to approximate the genuineness of calls in any BS.

Among the challenges associated with IoT, two challenges pose threats to the overall network connectivity, including battery life and the ability of edge devices to communicate over a long distance. One promising technology among low-power wide-area networks (LPWANs) is LoRa, which operates based on spread-spectrum modulation techniques. In [163] ML algorithms were adapted to edge devices to mitigate two challenges: life expectancy and the ability to communicate over long distances. To achieve this, LoRa is used for low-power transmission, and KNN is used for the activity classification process. They have accomplished an amazing low energy expenditure of 5.1 mJ in power consumption for activity classification, resulting in a battery life of 331 days. A similar technique can be deployed to IoT distributed edge devices to increase the device lifetime in a distributed manner.

Caching popular content at SBS for intelligence gathering in ultra-dense heterogeneous networks effectively decreases redundant data transmission and E2E delay. However, dealing with different data content is challenging and time-consuming. The authors of [164] used a clustering-based TDMA transmission scheme for content placement and user association. They used an offline training procedure using DNN to predict the user association for each cluster, where the input to the NN is the user channel gain and user demand. As the number of clusters increases exponentially as the number of users served by the SBS increases, the user cluster information and time duration of serving each cluster are optimized using the DNN. As a result, the user association to each cluster can be quickly identified, and the time required to obtain the optimal user cluster information is reduced. This method enables us to achieve efficient load balancing and user clustering in a centralized KDN.

(ii) *Knowledge derived from unsupervised learning*: As the number of connected devices in IoT for applications, such as smart cities, smart homes, farms, and factories, reaches more than 31 billion, providing secure communication and access control becomes a priority. In [165], the authors proposed a mechanism based on unsupervised clustering techniques to enable reasonable access control throughout the communication history of IoT networks. Their method is called INSTRUCT and has two separate algorithms for different types of traffic, one for TCP traffic and the other for UDP traffic. The proposed algorithm uses past communication data to allow access to IoT devices by installing new rules on the switches involved using the clustering technique. K-mean clustering was applied to distinguish between valid and invalid traffic captured by the switches. INSTRUCT achieves 100% classification accuracy for TCP traffic and 95% for UDP based on their comparison with signature-based manual analysis. This algorithm is suitable for a centralized KDN to provide access control to IoT devices.

According to [166], the authors used the K-mean algorithm for clustering SBSs based on location and traffic load. The proposed scheme has a dynamic cluster-based on/off switching mechanism for SBS. The clustering-based technique allows SBSs to coordinate their transmissions while reducing energy consumption and traffic load. This study formulated the problem as a noncooperative game between clusters, where clusters seek to minimize the cost function to reduce energy consumption. Based on the information regarding the location of SBSs and the capability of handling users and data traffic, the cluster determines their transmission power and on/off situation. The simulation results show improved overall performance when using the cluster-based coordination method in small-cell networks. The algorithm attempts to reduce the overhead on a centralized controller by allowing the SBS to decide based on their locally acquired information. Hence, this method can be fitted to the distributed architecture of the KDN.

As the demand for ubiquitous access to wireless data increases, Tabrizi et al. [167] proposed a clustering and spectrum assignment and resource allocation (CaSRA) to cluster nodes in hotspot densely populated areas. The K-mean clustering algorithm was adapted to maximize spectrum utilization and increase network performance. In their algorithm, a mobile device can act as a hotspot or slave. Once the mobile user is connected to the cellular network, it serves as a hotspot and provides broadband access to nearby devices known as slaves. The problem formulation is first to identify mobile devices that act as hotspots and then to obtain the users associated with each hotspot. To solve this problem, a modified version of K-mean clustering is used to cluster users based on their location and organize each cluster based on their maximum and minimum number of users. Then, the user with the minimum distance to both the BS and the center of the cluster is selected as the hotspot in that cluster. Further, using a graph coloring approach, power and spectrum are allocated to each cluster and from the cluster to hotspot and slaves. CaSRA increases the total number of supported users in a network with lower complexity. The centralized CaSRA scheme uses the BS to cluster the nodes into groups based on their distance, making the algorithm suitable for a centralized KDN.

In [168], they proposed a new multicarrier waveform classification for 5G communication systems. This study utilizes PCA and CNN on signal amplitude to mitigate noisy channels and obtain a high classification accuracy. Their algorithm can detect and cluster three multicarrier waveforms: universal filtered multicarrier (UFMC), filterbank-based multicarrier-offset quadrature amplitude modulation (FBMC-OQAM), and orthogonal frequency-division multiplexing-quadrature amplitude modulation (OFDM-QAM). Their method works even in a dense channel environment, where the transmission and detection of these three signals were not possible before. Moreover, compared to other methods, such as CNN using in-phase and quadrature (I/Q) modulation, it has better performance and less complexity. This technique can be used in a distributed manner to classify signals at 5G BSs. Hence, the scheme provides a distributed knowledge of signals.

6.2.3. User association

Current wireless communication networks rely on the existence of a cellular architecture. Cellular communication requires BSs for users to request, receive, and upload information. Every BS in the network has a coverage area that supports a specific geographic area with limited users. To increase the capacity of BSs in the cellular network, small cells were introduced to enable service providers to offload users from an overloaded BS to sub-BSs, namely macro BSs, pico BSs, and femto BSs. Therefore, more cellular networks are shifting toward heterogeneous networks (HetNets), enabling flexibility and low-cost deployment of new infrastructure. To associate users with an appropriate cell, the KDN requires reliable methods. Hence, some practical ML-based techniques have been investigated, including studies on SL and RL. Table 11 summarizes the studies surveyed in this section.

(i) *Knowledge derived from supervised learning*: One of the most promising technologies in 5G is the appearance of mmWaves to improve cellular capacity and increase overall performance. The mmWave systems bring different challenges to future networks, including the interconnectivity of several terminals with multiple BSs, and estimation of the user's CSI. Therefore, user association in mmWave networks is crucial for network durability and performance. The authors of [170] developed an ML-based user association for mmWave networks. They formulated the mmWave user association problem as a multi-label classification problem, where they adapted feature extraction from topological information and geographical location information. Specifically, users and BSs are categorized into samples and classes, where each sample in the network can be classified into multiple classes. Initially, each user in the network can be associated with multiple BSs, creating a multi-label classification problem. To avoid high dimensionality, multi-label classification methods are transferred into a series of single-label classification problems with lower dimensionality. Finally, after the training procedure and based on the information and features, the algorithm allocates samples to only one class. The research aims to make the user association without utilizing CSI, and the simulation results indicate high performance even with few training samples. Graphical model representation is utilized to solve the user association problem, suggesting a centralized KDN structure, although it can also be used as a distributed architecture.

Some of the applications of 5G are providing massive connectivity to vehicular users, including self-driving cars and UAVs. UAVs experience significant interference from different BSs on the ground with the ambiguity of which BS the user must connect to. To mitigate interference and maximize the performance of wireless links, Galkin et al. proposed an SL-based association scheme to associate UAVs with the most appropriate BS [171]. An NN is trained based on environmental information to identify the most suitable BS. UAVs' distances to BSs, received signal strength (RSS) from the BSs, and the location from which the interference comes from are fed to the NN. The role of the NN is to intelligently identify which BS provides the best channel quality. RF chains are utilized in their method with two separate antennas: a directional antenna and an omnidirectional antenna. An omni-antenna is used to measure the RSS from the nearby BSs, while a directional antenna is associated with the appropriate BS. The objective of this study was to maximize data transmission and channel capacity. The performance of the proposed NN algorithm is compared with the strongest-signal and closest-neighbor association schemes, where the simulation results demonstrate the superiority of the proposed method. Furthermore, the proposed NN can increase the probability of UAV coverage compared to non-NN BS association schemes. UAV users intelligently decide the BS they must associate with in the proposed study, hence a distributed user association scheme.

(ii) *Knowledge derived from reinforcement learning*: Cell range expansion is a technique to increase the coverage area of BSs to either support other users or to cover the blind spots, which increases the coverage area, network throughput, and cell-edge throughput. Expanding the coverage area can be achieved by adding a bias value to the user equipment (EU). Selecting an optimal bias value depends on various factors, including the radio resource ratio between the MBSs and pico-BSs. A dynamic method is presented to determine the bias value of each UE by using Q-learning algorithms [172]. Here, all the UEs learn the bias value independently with the aid of Q-learning to decrease the outage of UEs. The proposed Q-learning algorithm is a multi-agent learning system that allows every user to determine the bias value. The state of the system is the received signal power from both the MBS and pico-BS. The agent's action is to choose

Table 10
Knowledge-based strategies for clustering.

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[161]	Intelligent clustering based on trained supervised learning dataset	Centralized	Logistic regression	Traffic classification and balancing load	Extensive comparison among various classification methods
[162]	Automatic detection and drop of fraud activities	Centralized	Support vector machine and fuzzy clustering	Detecting the fraud communication activities	The proposed algorithm performs better compared to normal SVM
[163]	Efficient low power clustering solution for IoT devices	Distributed	K-nearest neighbor	Extending the battery life	The proposed scheme can reduce the transmission payload and lowering the transmission power
[164]	Intelligent clustering mechanism for ultra-dense heterogeneous networks	Centralized	Supervised deep neural network	Minimizes network sum energy consumption and reduces the transmission delay	The proposed algorithm is time efficient and has a 90% accuracy of user clustering
[169]	Smart clustering for random access networks	Centralized	K-mean clustering	Low convergence time	Machine type communication device association with conflict-free resource allocation
[166]	Dynamic switching on/off SBSs	Distributed	K-mean clustering	Increases the energy efficiency of SBSs	Clustering wireless small cell networks based on location and traffic load, but they did not consider the frequent mode transition
[167]	Intelligent node clustering in densely populated wireless areas	Centralized	K-mean clustering	Efficient offload traffic and reuse spectrum	The proposed algorithms jointly cluster the nodes and assigns spectrum and physical resources
[168]	Automatic multicarrier waveforms classification in future cellular networks	Distributed	Principle component analysis and CNN	Achieve low complexity and high accuracy	Classifying three waveforms including OFDM-QAM, FBMC-OQAM, and UFMC, but should be further tested in real channel environments with the existence of other noises

a bias value, and the reward of the system is determined by the BS when calculating the number of outages. The proposed algorithm increases the throughput and decreases the number of disconnected UEs. This algorithm is a greedy-based cell expansion by every UE, which can be used in a distributed KDN architecture.

In vehicular networks, continuous connectivity is key to network intelligence and automation. However, user association also plays an essential role in vehicular networks by providing load balancing between SBSs and MBSs. The authors of [173] proposed an online RL approach (ORLA) for network load balancing in VANETs. Their learning algorithm is divided into two main phases: initial RL procedure and history-based RL procedure. In the initial phase, the user and BS association problem is formulated as a multi-armed bandit problem. The vehicle association decision for each BS is the agent's action, and the reward is the network load balance. In vehicular networks, there are regularities in the spatial–temporal dimension due to urban traffic flows. In the history-based RL phase, by considering the dynamic changes of the environment and the spatial–temporal regularities, the association patterns obtained in the initial phase enable simultaneous load balancing of BSs. The history-based RL generates an association matrix for each BS based on the similarities between historical patterns and the current environment. The proposed algorithm is compared with the distributed dual decomposition optimization and max-SINR scheme, wherein ORLA outperformed load balancing in multiple cells. The algorithm provides a centralized architecture for future KDN-based networks.

Although information, such as channel quality, backhaul capacity, and SINR are important attributes of user association, the user's QoE has the same or higher priority. Pervez et al. [174] introduced a distributed user-centric backhaul-aware user association scheme via a fuzzy Q-learning algorithm to autonomously enable each BS/cell to use QoE and backhaul constraints to maximize the throughput of the network. Fuzzy Q-learning attempts to learn the optimal bias value (which is the action of the RL algorithm) by the fuzzy rules in an iterative manner by interacting with the environment. Specifically, users receive a bias value from different cells, guiding them to associate with the most suitable cell. At the same time, each bias value represents an attribute that satisfies the network performance metrics, including latency and throughput. The proposed algorithm achieves an optimal performance faster than Q-learning-based methods and, more importantly, improves the user's performance. Because users will have different requirements in the 5G system, each is associated with different cells based on their needs. Therefore, the above technique is best suited for distributed user association schemes in the KDN architecture.

Owing to changes in network characteristics and the increase in devices with different requirements, researchers have turned to HetNets. Even though it provides several benefits, it presents challenges, such as interference between SBSs, power control, and user association problems. The authors of [175] used a multi-agent deep Q-learning network (DQN) to solve some of these problems. Their algorithm consists of convex optimization and fractional programming, and uses DQN to jointly optimize user association and power management in OFDMA systems. In the Q-learning algorithm, the agent (or UE) must select the

Table 11
Knowledge-based strategies for user association.

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[170]	Trained sample is used for user association of the mmWave systems	Centralized	Supervised learning	Accurate user association	The proposed approach shows good performance even with few training samples and without CSI, but it consumes significant time in training process
[171]	Intelligent BSs selection by UAV users in 5G networks	Distributed	Supervised learning with neural network	Maximizes channel quality of the UAV-BS link	The proposed policy allows UAVs to select the suitable BS based on BS transmit power, distance and the location of the BS, can be extended with varying mobility rates
[172]	Efficient user association in heterogeneous networks after UE learning procedure	Distributed	Q-learning	Improving the system throughput and reducing the number of UEs' outages	Optimizes user cell association (PBSs or MBSs) through learning the cell range expansion
[173]	Online training approach and near-optimal association in dynamic vehicular environment	Centralized	Reinforcement learning	Efficient load balancing in the network	Adapting the user association policy via learning the spatial–temporal dimension regularities
[174]	Autonomous cell association in ultra-dense small cell networks by enabling user-devices to store learned values	Distributed	Fuzzy Q-learning	Improving the convergence time	The proposed scheme is memory-based user-centric backhaul-aware but may have an impact on user's memory usage
[175]	Intelligent adaptive decision making to maximize UEs' QoS in heterogeneous networks	Distributed	Multi-agent deep Q-learning network	Maximizes energy efficiency while jointly associated users	Joint optimization of user association and power management
[176]	Near-optimal solution to improve QoE in MEC-enabled live video streaming systems	Distributed	Deep reinforcement learning	Maximizing users' QoE	Efficient user association and resource allocation
[177]	Smart user association in symbiotic radio networks (SRNs)	Hybrid	Deep reinforcement learning	Achieving optimal user association policy without full real-time channel information	The proposed algorithm uses both centralized and distributed DRL approaches to make a decision for IoT devices, where centralized converges with few dataset and distributed is scalable

appropriate BS to create communication links and determine the transmission power. The transmission power of the user is the action of the agent throughout the learning procedure. The reward function is the sum energy efficiency of all the UEs. The objective of the learning algorithm is to maximize the expected accumulated reward under QoS constraints. The convergence of multi-agent DQN was analyzed in the simulation results, and it proved to be superior to traditional RL-based techniques. Moreover, the algorithm maximizes the long-term overall network performance and demonstrates efficient energy consumption. The distributed method above shows a solid connection to the distributed KDN architecture. Chou et al. used DRL to jointly solve user association and resource management problems in mobile edge computing (MEC) to improve the QoE for online video streaming in 5G networks [176]. The problem is formulated based on the Markov decision process (MDP) and analyzed by a deep deterministic policy gradient (DDPG) algorithm based on the supply demand interpretation of the Lagrange dual problems. First, they used the traditional optimization Lagrangian approach, where the source of the performance loss in this algorithm was identified as the Lagrangian multiplier update function. Then, they proposed a pricing function based on MDP for the update function, which was solved by DDPG. Here, the MDP is solved using a DRL, where the supply–demand inspires the agent's action as the output price of each video from the BS. Then, based on the prices for each video from different BSs, the UE is associated with a BS whose data rate is maximized. The reward function is defined as the sum of each UE's QoE, and the goal is to maximize it. Simulation results show that the proposed method achieves significant improvements in QoE, particularly in congested networks with low resources. This method can be used in a distributed knowledge plane for users to associate with a BS autonomously.

Symbiotic radio networks (SRNs) have been introduced to enable the coexistence of various networks and utilize resources globally and provide connectivity across multiple networks [178–180]. The authors of [177] focused on the symbiosis between IoT and cellular networks and the user association problems in SRN. They used the TDMA for cellular communication between users and BSs. Then, each IoT device is associated with one user to exchange information. The dynamic changes in the environment make the collection of real-time channel information difficult.

To overcome this problem, two DRL algorithms were utilized to guarantee optimal user association. One of the algorithms is centralized, which makes decisions for IoT devices based on globally available information. At the same time, the other is distributed and makes decisions based on locally available information. The DRL algorithm can have two states based on the proposed DRL algorithm. The first state is the action space in a centralized DRL-based user association, which is a matrix of cellular users with the associated IoT device. The second state is a distributed DRL-based user association scheme with one IoT device. The immediate reward in both schemes is the sum rate of all IoT devices. The proposed scheme shows optimal user association with high scalability, even when IoT devices increase in the network. This algorithm is suitable for hybrid KDN architectures because it takes advantage of both centralized and distributed algorithms.

6.2.4. Traffic and data aggregation

As networks are increasing in complexity and become more difficult to manage, embedding intelligence into devices will ease optimization, recommendation, organization, and management. Most studies in the networking area are distributed in nature, which makes it difficult to include ML-based algorithms for controlling the devices. KDN functionality provides an opportunity to bring intelligence and knowledge to the network. The KDN can collect global information to improve network performance. Traffic classification is a crucial activity in network management, and massive growth in Internet users has brought network traffic classification into attention. Table 12 summarizes the studies surveyed in this section.

(i) *Knowledge derived from supervised learning*: Raikar et al. integrated SDN architecture with SL techniques, specifically SVM, Naïve Bayas (NB), and the nearest centroid is used to classify the network data traffic [181]. First, in the learning phase of their algorithm, the training data are fed to the system to map the network traffic into defined classes. Later, real-time data were captured and mapped based on the trained SL for network traffic classification. In their method, the SDN controller utilizes three different SL algorithms to classify the data into HTTP, mail, and streaming. Their proposed solution was able to obtain high accuracy in all three learning algorithms with the highest accuracy for NB, followed by SVM and the nearest centroid. Their algorithm provides centralized data classification, which offers an opportunity to add intelligence to network devices.

(ii) *Knowledge derived from transfer learning*: To avoid training data from scratch, researchers in [182] method to address multi-class traffic classification problems, and they utilized Maxent as the base classifier in their approach. A new classification task in TrAdaBoost was used to extract labeled data from several network traffic sources. Next, the Maxent model is used to classify and convey traffic knowledge from the source domain to the target domain. The proposed scheme was trained and transferred as prior knowledge for different environments to reveal its performance. They tested their method with two traditional ML algorithms based on Maxnet, known as NoTL and NoTL advance, where TrAdaBoost achieves better performance compared to the other two methods. Their learning algorithm can achieve high accuracy in classifying network data traffic and provide a promising solution for centralized KDN architecture.

6.3. Mobility management

Another critical aspect of service delivery in wireless networks is mobility management. Mobility management helps to develop user mobility and handover predictions. In future wireless networks, mobility management is crucial for network automation. Knowing the location of users and their activities can help generate intelligence to increase network performance. Recently, there have been many advancements in mobility prediction and handover management in the field of ML. In the following sections, some ML techniques for the two essential components of mobility management are presented.

6.3.1. Mobility prediction

Movement is an inherent nature of mobile devices, and predicting the next location of these devices is called mobility prediction. Knowing the next location of the user can improve the network performance, especially in resource management, D2D, and V2V communication [187]. In this section, some promising methods using ML algorithms are presented. These methods provide a huge advantage in the KDN architecture for predicting user mobility patterns in the next generation of cellular networks. Table 13 summarizes the studies surveyed in this section.

(i) *Knowledge derived from supervised learning*: Learning and knowing the next location of mobile equipment enables mobile applications and a coherent handover process. Location prediction techniques use the historical trajectory information of mobile users to guess an individual's next position. Many studies have introduced location prediction methods with acceptable accuracies [188–190]. However, some of them perform erratically when a user's activities change in a new area, while others encounter the "cold start" problem when the user's trajectory information is sparse. The authors of [183] involved the user's activity patterns and historical data and proposed an SL-based location prediction method. Their technique evaluates the next activity of the mobile user by modeling individual activity patterns rather than directly predicting the next location of the mobile user. Then, it predicts the next location of the user based on the obtained next activity. Using real-life GPS trajectory data, the simulation results show a smooth upgrade and robust performance based on the proposed prediction procedure. Having knowledge of the user's activity and location in a centralized KDN architecture improves the network performance, especially when a controller needs to decide on resource allocation and cluster head selection.

(ii) *Knowledge derived from unsupervised learning*: Owing to the dramatic increase in users across networks, configuration and optimization have become more complicated. More users push the network to split into small cells and SBSs. More frequent handovers occur once there are more SBS in the network, and because of the complexity of the indoor environment, this task is more delicate in an indoor scenario. Self-organizing networks (SONs) are the key to the next generation of mobile networks for self-healing, self-optimization, and self-organization. Sinclair et al. proposed a modified self-organizing map (SOM) method to make indoor location predictions of users while a handover request occurs [184]. Their method determines whether the indoor user should be connected to another BS or prohibit the handover based on their location information. SON utilizes an unsupervised NN that allows the learning algorithm to generate a low-dimensional output space from high-dimensional input data. The input data depend on the mobile terminal approach, whereas in this scheme, the angle-of-arrival (AoA) of the user and the reference signal received power (RSRP) are fed to the NN. Based on this information, the SOM is capable of estimating the user's actual physical location. Accordingly, based on the pre-determined zones, which correspond to prohibited and permitted areas, a handover decision is made. The simulation results demonstrate that the proposed algorithm reduces the total number of handovers by 70% while allowing the necessary handovers to proceed. The proposed algorithm has an automatic system that uses monitoring, analyzing, planning,

Table 12
Knowledge-based strategies for traffic and data aggregation.

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[181]	Fully automated management in SDN-based networks	Centralized	Supervised learning methods	More than 90% accuracy in data traffic classification	Provides security monitoring, fault detection and traffic engineering
[182]	Real knowledge transformation from one domain to the target domain for efficient traffic classification	Centralized	Transfer learning	Achieving high classification accuracy	In contrast to other studies training data and test data have different distributions

Table 13
Knowledge-based strategies for mobility prediction.

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[183]	Collected user activity can be used to predict both location and the individual's activity simultaneously	Centralized	Supervised learning	Robust performance and smooth prediction upgrade	Predicts the users' next location by modeling users' activity patterns from GPS real-life trajectory data
[184]	Self-organizing networks for handover management in LTE femtocell BS networks	Centralized	Unsupervised learning	70% reduction in unnecessary handovers	Handover management based on the user indoor location
[185]	Allowing BSs in LTE heterogeneous networks to automatically and autonomously discover the RF conditions in their cell edge	Centralized	Unsupervised shapelets	Clustering result with an average accuracy of 95%	Classifies the users' trajectories and optimizes handover parameters
[186]	Smart mobility prediction for efficient service migration in the mobile service provision problems	Centralized	Deep reinforcement learning	Offloading traffic and reducing latency in the system	Optimizes the service provision problems in mobile edge computing

and execution phases to perform a centralized knowledge-based concept. Moreover, to allow BSs to autonomously discover the RF conditions at the cell edge and their impact on the handover parameters, unsupervised-shapelets and data mining techniques were proposed to recognize patterns in the RSRP measurement reports from users [185]. Their method makes position estimation once a handover is triggered. Based on the positioning, the BS discovers new patterns while the network characteristics change and calculates the number of clusters in the network. The simulation results illustrate that even without prior knowledge, the algorithm provides 95% accuracy in clustering the nodes and predicting the user's location. The proposed algorithm uses cluster heads to identify user's movements and trajectories. This method is suitable for a centralized KDN for user-trajectory prediction.

(iii) *Knowledge derived from reinforcement learning*: In [186], a mobility prediction model based on DRL at the edge of the network was proposed for mobile users. They designed a DRL framework to offload traffic by training a DQN for mobility prediction. Their method comprises a glimpse mobility prediction model that gathers users' mobility patterns and trains them in the DRL. The algorithm first assumes that the controller can select the best edge server and apply the DRL. Then, the controller predicts the users' future locations based on historical data and past user mobility using the DRL agent. The authors used the actual human trajectory and user latency to obtain the performance of their algorithm. The experimental results show that the glimpse mobility model outperforms the perfect mobility model and chooses the lowest latency service request. Therefore, it is essential to have a similar model at the edge network to allow the controller to select the best strategy with the least latency. Additionally, mobile service providers can determine the expected latency experienced by users to receive data packets in future cellular networks. The low-latency ML-based algorithm in this work suggests a centralized knowledge plane to predict the series of locations and timing of the users.

6.3.2. *Handover management*

Recently, the deployment of SBS has contributed to increasing network performance to provide acceptable QoE. However, deploying SBSs means that users will have more frequent handovers, which can negatively impact users' QoE [191]. In the upcoming section, some of the studies on handover management based on ML techniques are presented to assist future KDN networks. Table 14 summarizes the studies surveyed in this section.

(i) *Knowledge derived from supervised learning*: Handover in the context of wireless communication means passing the control of a UE from its serving eNB to the next nearest eNB without any interruption. With today's continuous connectivity of users with a high demand for data rate and low latency, handover management is becoming more complex. Additionally, to provide seamless connectivity to users, a handover is required. However, challenges such as security and QoS are expected. Many

Table 14
Knowledge-based strategies for handover management.

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[192]	Smart handover management solution for LTE networks	Centralized	Supervised learning with neural network	Improves users' QoE	The handover algorithm provides better QoE by choosing the appropriate cell
[193]	Intelligent preparation decision for better handover in 5G mmWave networks	Centralized	Supervised learning with deep neural network	Reduces the signaling overhead while improving the success rate	The proposed prediction-based conditional handover demonstrates 98.8% and above accuracy of prediction
[194]	Constructs a centralized knowledge-based server for future wireless networks	Centralized	Supervised learning with recurrent neural network	90% traffic forecasting accuracy	The proposed model utilizes an artificial intelligence into the mobile network for adaptive handover optimization
[195]	Efficient handover management to support intelligent vehicular networks	Centralized	Supervised learning with recurrent neural network	Accurate prediction of handover trigger decision	The proposed algorithm initiates early handover registration to mitigate communication disruption
[196]	Dynamic mobility management in small cell 5G networks	Centralized	Fuzzy Q-learning	Reduces the unnecessary handovers while keeping an acceptable call dropped ratio	Optimization of handover parameters for different UE speeds to maximize network efficiency
[197]	Smart user-centered handover decision making for an open-access femtocell networks	Centralized	Q-learning	Increases network capacity with less number of overhead	The proposed scheme observes the historical data and predicts the future user association, but the model only deploys one MBS
[198]	Adaptive user selection and cell association in open-access femtocell networks	Centralized	Fuzzy Q-learning	Fewer number of handovers with opportunistic channel capacity	Adapting the learning algorithm from the past and future experiences
[199]	Optimal handover management in large-scale wireless mobile networks including IoT	Centralized	Deep reinforcement learning and supervised learning	Reduces the handover and ensures the system throughput	SL uses the traditional handover parameters to initialize RL, then RL is used to obtain the optimal controller for each UE
[200]	Robust tuning of handover parameters and load balancing in LTE networks	Centralized	Fuzzy Q-learning	Traffic engineering	Self-organizing mechanism for joint optimization of load balancing and handover management
[201]	Intelligent handover management for LTE-advance networks	Centralized	Q-learning	Minimizes the handover failure rate and handover ping-pong effect	The proposed algorithm selects the most efficient eNB using reference signal received power, reference signal received quality, and other criteria
[202]	Simultaneous monitoring of signal strength and optimal hand over control in 5G networks	Centralized	Reinforcement learning	Maximizes the network throughput	A centralized RL agent uses the radio measurement reports from the UEs for handover optimization
[203]	Optimal decision-making for predicting handover in mmWave networks	Centralized	Q-learning	Maximizes the throughput	The proposed scheme finds the optimal policy based on the pedestrian information including the location and velocity
[204]	Intelligent handover learning in mmWave networks	Centralized	Double deep reinforcement learning	Improves the QoS	The proposed framework provides the optimal BS selection policy
[205]	Self-tuning of handover parameters and power allocation in heterogeneous networks	Centralized	Deep reinforcement learning	Maximizes the throughput and reducing the handover frequency	The multi-agent algorithm train decentralized policies for each UE to achieve better performance

studies have used ML techniques for self-organizing networks to improve network performance. Ali et al. presented an SL-based handover management scheme to improve the QoE for LTE users [192]. They utilized historical data to learn how the QoE of users changed when the handover decision was made. In particular, eNB gathers measurement parameters from the users, including the user's radio link condition of the current eNB, the user's neighboring eNBs, and the user QoE resulting from past experiences once handover was made. This information is fed to the two-level NN model, where the first level determines the QoE in terms of complete download or incomplete download, and the second level is trained to approximate the file download time. Based on the handover algorithm, the current serving eNB triggers the handover to the next eNB with uninterrupted service to the user. Their algorithm assigns data for users to download and measures the amount of information lost due to handover. The simulation results show that almost 96% of the data were downloaded even when handover occurred. BSs can use the proposed method to create a centralized KP and perform handover management with a low data loss ratio.

Owing to the vulnerability of mmWaves in 5G, which can cause sudden changes in the received signal strength, handover can often be misleading and lead to bad decision making. One of the contributions of 5G networks is the conditional handover (CHO). To enhance the performance of CHO, Lee et al. proposed a novel prediction-based CHO (PCHO) scheme based on a DNN [193]. Their method uses the former blockage information to predict the best gNB. Here, the signaling patterns from the BSs are collected and used in the algorithm. In this study, the authors focused on the preparation phase of handover because they claim that the preparation phase is the most vulnerable period in handover. The preparation phase occurs when the signal quality experienced by the UEs is low, and the interference from other BSs is severe. In fact, it is most likely that the UE experiences plenty of handover message delivery failures in the preparation phase. Therefore, to achieve accurate handovers, a DNN is trained offline via a training dataset, allowing the current gNB to make a real-time decision on the preparation period to predict the next gNB. PCHO outperforms most of the CHO schemes in terms of the preparation success rate and prediction accuracy of the new BS for handover. The proposed algorithm also decreases the signaling overhead compared to the current studies on CHO-based mechanisms. The recommended scheme is suitable for centralized BS to manage handovers. In [194], an RNN was used to optimize the handover in 5G networks. In this work, mobile users increase the transmission of useful information on wireless channels and reduce service information. This is because each device is equipped with an NN that helps to process sensitive data locally. The result of the NN is always sent back to the knowledge-based server for further processing or storage. Their technique reduces the amount of service traffic transmitted by users through the communication channel and self-organizes the handovers in heterogeneous networks. The proposed method is simple, reduces system complexity, and provides sufficient knowledge for a centralized KP.

Intelligent vehicular networks (IVNs) have attracted many researchers because of their real-time road safety services and other essential applications for vehicles. However, the development of efficient and robust wireless communication in vehicular systems is still challenging for content delivery. This is mainly because of the high mobility rate of vehicles that disrupt connectivity. The authors of [195] proposed a two-tier ML-based scheme for intelligent handover management in an IVN. In the first tier, an RNN model is used to predict the receiving signal strength of APs to make a handover trigger decision. A stochastic Markov model is utilized in the second tier to select the next AP by considering the vehicle flow projection. The handover trigger was divided into three parts: data processing, learning phase, and prediction phase. They used an offline ML technique to model a long short-term memory (LSTM) network (which is an RNN used for time-series sequence prediction [206]) to predict the RSSI of the AP. Based on the acquired RSSI values, the system decides whether to trigger a handover. This decision shows if the handover must occur before its actual execution, which initiates an early handover registration process to avoid any disconnection of the signal while switching between APs. The proposed scheme outperformed the related models in terms of prediction accuracy. The proposed method is suitable for future self-driving vehicles to avoid collisions and can receive information at any instance of time. As a result, a centralized KP is needed to collect all the information and prepare to take handover actions when necessary.

(ii) *Knowledge derived from reinforcement learning*: The authors of [196] proposed a fuzzy Q-learning algorithm for two-tier networks that include macrocells and small cells. Their method uses the call drop rate and signaling overhead caused by the handover to formulate the system state in the Q-learning process. The Q-learning action is based on the possible values of the trained handover margins. The proposed algorithm aims to obtain a trade-off between user experience disruption by call dropped ratio and signaling overhead. Simulation results illustrate that the framework reduces the number of handovers while maintaining an acceptable call drop ratio. In [197] the authors adapted RL techniques to find the best user association in a nonstationary femtocell network. They proposed a framework in which specific cells explore past cells' behavior and use Q-learning to predict the future state. The simulation results show growth in capacity and decrease in unnecessary handovers. Moreover, the authors of [198] presented a user association technique to avoid handover for situations with high expectations of communication disruption, such as dense environments. This algorithm is similar to the work above, where the learning algorithm predicts the state of the network based on past experiences. They used a fuzzy inference system (FIS) to facilitate Q-learning parameters during the training process. In this manner, their algorithm achieves better performance when the environment changes dynamically. All the proposed algorithms can provide knowledge for a centralized KDN to overcome handover problems in small-cell networks.

In [199], a two-layer framework for handover control in large-scale wireless networks was introduced. In the proposed framework, a centralized controller clusters UEs based on the mobility patterns and utilizes an asynchronous multiuser DRL scheme to control the handover. DRL uses the actor–critic learning method to achieve an optimal policy for handover in each cluster. The algorithm uses online and offline methods, wherein the online method UEs keep processing and fetching the weight parameters periodically, while in the offline method, UEs behave as static controllers. Consequently, there are two different controllers in the network: one managing the mobile devices and the other managing the controller (main centralized controller). Their work is promising for future wire-less networks as it acts efficiently with the dynamic structure of the network and can train fast even with newly arriving UEs. Therefore, adapting such a system at the KP is essential to address scalability issues and control handovers more robustly.

As a result of the growing complexity of the network, there has been substantial research interest in SON. 3GPP has defined some of the principles and concepts for automating the network configuration and management for improving the network [207, 208]. In this context, industry and academia have defined load balancing and handover optimization as the most important parameters for self-organizing mechanisms in RANs. The authors of [200] proposed a fuzzy logic and RL technique to jointly improve the handover and load balancing for LTE users. The fuzzy logic tunes the handover parameters at the cell adjacency level. The Q-learning algorithm was used to optimize the fuzzy system to improve the network performance by forcing the scheme to select the most appropriate handover action. The RL action chooses the BS while handover occurs, and the reward function is based on jointly maximizing the handover management policy and load balancing function. The proposed algorithm proves that the network performance increases as both entities (handover and network load) run together to optimize the performance compared to schemes where one entity is running on SON. 3GPP introduced LTE-A/LTE-advanced to improve network coverage, capacity, and data rate, which provided low latency and a high data rate. The authors of [201] utilized the AHP-TOPSIS method with Q-learning algorithms to make intelligent handover management. The proposed method selects the optimal eNB within an appropriate trigger time. To choose the most suitable eNB for handover, the algorithm uses RSRP, the reference signal received quality (RSRQ), uplink signal-to-interference plus noise ratio, location, moving direction of the UE, and current load on the eNB. Based on the running application on the network, such as speed-sensitive or delay-sensitive, AHP-TOPSIS is associated with UEs. According to the obtained ranks, UEs are allocated to a new eNB, where Q-learning evaluates the optimal triggering point for handover. The proposed algorithm is evaluated by numerical and simulation results, proving that their technique minimizes the handover failure rate and handover Ping-Pong effect compared to the fuzzy multiple-criteria cell selection (FM-CCS) scheme and other prior methods. Both algorithms can be deployed in a centralized KDN for future cellular networks.

Handover is more challenging in 5G and beyond networks owing to the high attenuation rate of mmWave signals. Non-line-of-sight (NLOS) signals caused by random obstacles, rain, etc., create a limited coverage range in mmWave networks. Overcoming this challenge requires deploying several small cells, which will cause frequent switching of user connections between the BSs. The authors of [202]used a centralized RL-based algorithm for handover optimization in 5G cellular networks. Their method chooses an appropriate BS based on radio measurements obtained from the user. Handover management is formulated as a sub-class of RL contextual multi-arm bandit (CMAB) problems. The BS collects the RSRP reports from the UE and forwards them to a centralized CMAB where the handover decision takes place, which is the action of the learning agent. The agent's reward is a function of the link beam of the RSRP, which is proportional to the network throughput, and the aim is to maximize the throughput. The proposed mechanism performs optimally in many simulated environments, demonstrating superiority compared to the results obtained with state-of-the-art algorithms. Similarly, the authors of [203] used RL techniques to predict the optimal handover decision-making action based on the information acquired by pedestrian movements. They utilized the location and mobility rate of pedestrians to learn the optimal handover policy to maximize the throughput of the network. Their network consists of a single station, mmWave APs, an access controller, and a human tracking module. The human tracking module uses an RGB camera to collect the position and velocity of the users. Based on the collected data, the access controller performs a Q-learning algorithm and makes a handover decision. The action of the learning algorithm in each iteration involves choosing one of the deployed APs. The reward function performs the data rate of the communication link once a handover is made. The aim is to maximize the reward in every iteration based on the algorithm observation. The numerical results demonstrate the superiority of the Q-learning-based handover policy prediction compared to the existing heuristic handover decision-making in terms of achieving higher throughput. Consequently, the proposed algorithm for solving handover problems in mmWave systems can be used in a centralized KDN architecture.

For 5G technology, Mollel et al. proposed a novel offline scheme based on the double-DRL or DDRL algorithm to minimize the frequency of switching between BSs [204]. Their algorithm intelligently makes handover decisions by maximizing the user-BS connection duration in a centralized architecture. To obtain the optimal BS, a trajectory-aware handover optimization scheme is introduced. This method uses the received SNR values from different BSs at any time, and then this information is used to map the user's exact location. In this way, they also considered obstacles between users and BSs, including trees, building vehicles, and people (using wireless insite software (WI)). Specifically, the offline learning framework simulates the environment and collects the data. Then, the DDRL agent trains and develops a model accordingly. The action in this learning problem is the selection of BSs. The reward is a function of minimizing the number of handovers and maximizing the average throughput of the network. The simulation results show robust performance while outperforming existing ML-based conventional schemes.

Guo et al. studied handover management and power control problems in a two-tier heterogeneous network consisting of mmWave and SBSs [205]. Specifically, they modeled the handover management and power controller problems as a fully cooperative multi-agent RL (MARL) algorithm via the proximal policy optimization (PPO) technique. They introduced a centralized training procedure, which collects global information from UEs for training purposes, with a decentralized execution policy sent to each UE after the training phase. In particular, the aim is to utilize DRL for handovers optimization to maximize the overall network throughput and reduce the handover frequency. The action of the UE involves the BS selection and power tuning. The reward of the training process is a function of throughput and handover penalty. The proposed algorithm formulated a decentralized policy based on multi-agent PPO (MAPPO), which increases the throughput and reduces the handover frequency.

6.4. Localization

Many applications depend on high localization precision to provide the best service quality. Localization is divided into two main categories: indoor and outdoor. A global positioning system (GPS) has been widely used for outdoor localization. However, indoor positioning with GPS is not applicable because of the complexity of the indoor environment. Signal attenuation is very high in indoor environments owing to condensed materials and obstacles. Therefore, indoor positioning methods, such as time of arrival (TOA), time difference of arrival (TDOA), received signal strength (RSS), etc., are used for location approximation in closed areas. However, more recently, researchers have started adapting ML algorithms to increase the accuracy of indoor localization in wireless networks. This section presents an overview of ML techniques deployed in indoor localization. Most mobile device activities occur indoors, which means it is essential for the 6G cellular network to have a near-exact location of the user before providing any services. KDN can learn from the historical data of a user and predict the user's location. Hence, more reliable services can be offered by KDN-based networks. Table 15 summarizes the studies surveyed in this section.

Table 15
Knowledge-based strategies for indoor localization.

Article	Knowledge objective	Architecture	ML algorithm	Deliverable	Conclusion
[209]	Adaptive tuning to maximize indoor localization	Distributed	K-nearest neighbor	Achieves location error as low at 1.7 m	The proposed algorithm uses RSS-level-based feature scaling model to improve the accuracy of indoor localization
[210]	Efficient near real-time localization system for navigation and monitoring in mobile devices	Distributed	Support vector machine	Significant improvement in convergence time and prediction accuracy	The trained offline scheme can decrease the online learning error by 0.8 m
[211]	Robust and efficient wireless localization method	Hybrid	Relevance vector machine	Reduces the computational complexity while achieving acceptable localization accuracy	The proposed algorithm uses a classifier to identify NLOS signals
[212]	Optimal indoor localization for UWB networks	Distributed	Support vector machine	Significant performance improvement in various practical scenarios	The proposed algorithm estimate the indoor localization ranging error to improve the ranging error mitigation
[213]	Optimal K-nearest neighbor positioning algorithm for WLAN	Distributed	K-nearest neighbor	Maximizes the accuracy with optimal number of reference points	Adaptive indoor positioning system
[214]	Introducing a novel signal fingerprint for geographical localization in LTE networks	Distributed	Neural network	Reduces the calculation time	By only using one LTE eNB, the algorithm measures the location with a maximum error margin of 6 meters
[215]	The trained 3D model is used for accurate indoor localization	Distributed	Deep autoencoder network	High-performance 3D localization in large indoor spaces	The proposed method extracts RSS measurements to increase the indoor positioning accuracy
[216]	Online real-time positioning for mobile devices based on the offline trained data	Distributed	Kernel principal component analysis (KPCA)	achieving 2 m positioning error while reducing the size of the radio map	The algorithm uses spatial division technique based on Random Forest technique to increase the accuracy of localization
[217]	Blind indoor localization using minimum information about the surrounding	Distributed	Principal component analysis	Accurate trajectory estimation	Ability to distinguish floors in a building and trajectory learning
[218]	High-performance localization system with smart adaptation mechanism	Distributed	Autoencoder	Achieves high accuracy without utilizing radio path loss model	The proposed scheme outperforms maximum likelihood estimation, fingerprinting methods and generalized regression NN
[219]	Suboptimal solution for indoor location-based services	Distributed	Deep learning	Achieves high positioning accuracy	The algorithm utilizes CSI phase information for fingerprinting and outperforms three benchmark schemes based on RSS or CSI
[220]	Optimal localization algorithm for mobile devices	Distributed	Deep autoencoder network	Maximizes the localization accuracy	The algorithm ensures acceptable error ranging by using CSI and estimated AoAs

(i) *Knowledge derived from supervised learning*: In [209], the authors presented a feature-scaling-based KNN (FS-KNN) to improve the localization accuracy. The algorithm depends on the measured RSS reported by the MS, which accounts for the actual relationship between the signal differences and geometrical distances. To obtain the parameters of the weight function, iterative training was established to tune the parameters. After training the model, the algorithm finds the optimal values corresponding to the actual distance between a newly received RSS vector and each fingerprint. Then, the user location with an average error as low as 1.70 m is determined by solving a weight mean of locations based on K nearest reference points. This algorithm has two phases, which train the system in the offline phase and use it for online location estimation. The offline trained algorithm can be used as knowledge to estimate the user location in an indoor environment with acceptable accuracy. The RSS information is gathered by APs across the network, which is then processed by MSs in a distributed manner for location estimation.

A real-time, precise, and reliable localization system can determine the acceptable position of any portable device with opportunities for tracking objects/people, navigation systems, monitoring devices, and other location-based services. Interest in ambient intelligence, which enables people or systems to be aware of the user's presence, is increasing [221–223]. The main issue in ambient intelligence is determining the position of the user with high accuracy. As a result, there are ongoing investigations based on indoor localization to propose new algorithms to improve the accuracy of indoor positioning systems. The authors of [210] proposed an online independent support vector machine (OISVM) for indoor localization that avoids training from scratch. Their model uses the RSS of WiFi signals to make online predictions and facilitate mobile devices. The proposed model includes two phases: offline and online. The algorithms learn through pre-collected RSS with reference point (RP) labels appended to the corresponding RSS during the offline phase. The offline phase also incorporates kernel parameter selection and data sampling to deal with the imbalanced dispersion of the data samples. In the online phase, new RSS samples are collected by a centralized local AP for online learning and to estimate the location. Compared to traditional SVM methods, their method can balance the accuracy of localization and model size. From the simulation results, the location estimation error decreased by 0.8 m, while the training phase time and period were reduced considerably compared to the traditional techniques. The proposed technique can be used via a distributed architecture of KDN.

Due to the complexity of the indoor environment and the existence of various obstacles, high-accuracy localization prediction is rarely achievable. This is due to the NLOS radio blockage and insufficient information from the nodes. To this end, it is helpful to identify and categorize the NLOS signals. Nguyen et al. developed a robust relevance vector machine (RVM) for ultra-wideband (UWB) signals using time-of-arrival (TOA) [211]. They introduced a hybrid two-step iterative (TSI)-based localization algorithm. In particular, they utilized cooperative localization, meaning that both centralized and distributed approaches for positioning contributed to estimating the location. In centralized localization, a central processor gathers the node's information and builds a map. This map is later sent to nodes, and the nodes perform location estimation in a distributed manner. Additionally, they designed a classifier and regressor to mitigate the limitations of SVM. The RVM classifier can classify the line-of-sight (LOS) and NLOS signals received from the unknown node positions. At the same time, an RVM regressor was adapted to predict the ranging error of the input data. First, the RVM regressor is trained using a feature vector as input data, which consists of maximum amplitude, received energy, mean excess delay, and so on. The benefits of the RVM classifier include the use of a smaller number of relevance vectors than the support vectors in the SVM and mitigating the range estimation error. This method proved to reduce the communication overhead and computational complexity while providing high localization accuracy. In contrast to this work, the authors of [212] presented a novel approach to mitigate the ranging error by directly mitigating the bias occurring in both LOS and NLOS. Therefore, explicit signal identification between LOS and NLOS is not necessary. Specifically, an SVM-based two-class non-parametric regressors method is used to learn and map the features extracted from the received signal to the ranging errors. The simulation results demonstrate that the proposed regressors greatly improve the performance in various environmental scenarios compared to conventional methods. Both studies presented above use UWB anchor nodes to estimate the location of objects, where UWB nodes collect the surrounding information and process it in a distributed manner to compute the location.

In addition to SVM and RVM, some researchers have used the KNN to achieve acceptable indoor localization. For instance, Xu et al. [213] proposed an optimal KNN positioning algorithm based on theoretical accuracy criteria (TAC) in WLAN indoor environments. In this method, the optimal number of nearest RPs that can locate the user is theoretically analyzed. The KNN-based localization algorithm demonstrated that even with k=1 and k=2, outperforms other settings for static localization. The proposed algorithm was verified in various scenarios, such as office rooms, static positioning, straight corridors, and dynamic tracking situations. Another promising method that helps localization is the adaptation of the NN. The authors of [214] developed a unique mapping between the characteristics of radio channels and geographic locations. A feature extraction algorithm is used to select the channel parameters with non-redundant information that is formulated as a fingerprint vector. The channel parameters were taken from the LTE downlink signals. A feedforward NN is utilized to input the fingerprint data and output the UE locations. The NN is trained and used by the UE to estimate their location in the network. The proposed algorithm was investigated for both indoor and outdoor environments, and experimental results demonstrate that by using only one LTE eNB, this technique reaches a median error distance of 6 m and 75 m in indoor and outdoor environments, respectively. Both studies proposed above are suitable for a distributed architecture of KDN [213,214], where UE uses the information coming from the BS to calibrate their location. Both algorithms provide a fast and accurate location-based technique, which is promising for the future architecture of the KDN.

(ii) *Knowledge derived from unsupervised learning*: To increase the positioning accuracy, the authors of [215] introduced a novel denoising autoencoder via Bluetooth low energy (BLE)-based indoor localization (DABIL) method. In this study, the autoencoder extracts useful fingerprint patterns hidden in the RSS indicator measurements and constructs a fingerprint database to show the reference locations in 3-D space. The fingerprint patterns were built offline using raw RSSI values from different beacon stations. RSSI measurements of all the nearby BLE beacon stations were collected and used in the training phase and further denoising autoencoder for fingerprint construction. After training, the autoencoder is used as a trained fingerprint to estimate the probability of the target being located in a reference location. Finally, the KNN algorithm was used to estimate the location of the target. Field experiments demonstrate that 3- D fingerprinting can significantly increase the localization accuracy. Additionally, their method performs better in terms of horizontal and vertical accuracy than deep learning-based algorithms and conventional fingerprinting methods. Their algorithm can be utilized in a distributed KP to perform location-based estimation. To reduce the computational complexity, another fingerprinting-based localization system was introduced in [216]. They proposed a dimension reduction method that uses maximum likelihood estimation and utilizes kernel principal component analysis (KPCA) for feature extraction. To reduce the storage space, KPCA divides the fingerprinting radio map into sub-regions. The proposed indoor localization can reduce the size of the radio map by 74% while achieving a 2 m positioning error. The algorithm is first connected to WiFi to upload the RSS information using devices for centralized processing. Then, the information is sent back to the devices to make the location estimation in a distributed manner. Furthermore, PCA is combined with linear discriminant analysis (LDA) to extract lower-dimensional features from WiFi RSS signals in [217]. The algorithm provides floor classification, landmark detection, and trajectory of users. The experimental results from different levels of office floors show that the positioning accuracy is improved. To deal with the unpredictability of wireless channels, Dai et al. employed a multi-layer neural network (MLNN) for RSS-based indoor localization problems without considering path loss or comparing the radio map [218]. The MLNN includes three main layers/sections: transforming data, denoising, and locating sections. In the transforming section, the RSS of an unknown node is taken as an input and maps to the corresponding hidden layer through a deterministic mapping function. In the denoising section, all the nodes' RSS errors are improved by using a multi-layer denoising architecture. Finally, from the location section, the corresponding square grid labels were detected to estimate the location. They have a two-stage training procedure, where the first stage uses an UL algorithm for pre-training each layer and a fine-tuning stage to minimize the error of the entire network. The proposed algorithm shows higher location accuracy than the maximum likelihood estimation (MLE), generalized regression neural network (GRNN), and fingerprinting methods. The proposed works in [217,218] suggest a distributed KDN architecture.

Indoor fingerprinting localization systems that use RSS-based WiFi measurements are easy to implement and have low hardware requirements. However, RSS-based methods have two main problems. First, the RSS values are highly random and can change according to path loss, shadowing, etc. Second, the RSS formation is the average of all the obtained received signal amplitudes. To tackle this problem, the authors of [219] proposed a novel deep-learning-based indoor fingerprinting system via CSI. The CSI amplitude responses are fed to the input of the DNN with a greedy behavior to reduce the complexity of the training phase. The weights in the DNN are stored as fingerprints to aid the localization procedure in the online phase. The online process uses probabilistic data fusion based on the radial basis function (RBF) to estimate the user locations. Their algorithm was examined in varying propagation environments and showed promising results with high accuracy. The same group of authors of [220] considered the calibrated CSI data of the 5-GHz OFDM channel information for indoor fingerprinting. Similar to other methods, their algorithm consists of two phases: online and offline. In the offline phase, a deep autoencoder network is utilized for each user position to rebuild the CSI phase information, where the weights are stored as the fingerprint. In the online phase, once a new user location needs to be obtained, the algorithm uses a probabilistic method that uses a weighted average of all the reference locations in the fingerprint to estimate the position. The proposed algorithm was validated with experimental results, which indicated that the proposed method was superior to the traditional CSI and RSS-based methods. Proposed studies above establish localization in mobile devices in a distributed manner.

7. Terms and conditions associated with machine learning algorithms for generating knowledge

In this section, the terms and conditions associated with KDN problems are generalized for researchers to consider before using knowledge in the network. These conditions are categorized to identify the type of problem KDN is required to solve: implementation complexity, time consumption, training data, and the differences between ML techniques in the same KDN problem. After checking each condition and meeting the requirements, a final decision can be made on whether to adapt the KDN algorithm and which ML algorithm is more cost-efficient.

7.1. Identifying suitable ML algorithm

The first and most crucial step is to discover the type of ML algorithm that is most suitable for any particular wireless communication problem. The majority of wireless communication problems are solved within a few different ML algorithms, categorized as regression problems, classification problems, clustering problems, and MDP problems. In regression problems, the ML algorithm is required to predict a continuous value output given an input. In classification problems, the ML algorithm needs to predict a discrete value output, usually answered by a yes or no, and zero or one to essentially identify the class to which the input belongs. The clustering problems are ML techniques, where the data are grouped based on their type or value. Finally, MDP problems are ML techniques that require taking action in the current system state based on the feedback reward resulting from the previous action. Moreover, all the above ML problems may involve feature extraction, which means that one or more inputs affect the output value. Feature extraction for any problem is performed manually or algorithmically. Consequently, for any wireless communication problem in the above categories, KDN can use an ML technique as a possible solution and derive knowledge from them. For instance, to solve caching problems in wireless networks, one solution is to acquire the content request probabilities from users in the network. This problem can adopt historical data and associate the model with regression problems. It takes the user profile as an input and then predicts the user's content request probabilities as the output. On the other hand, the same problem can be solved by MDP, where, in every state, the agent can predict the associated cache content request. Furthermore, in user association problems, clustering algorithms are used, and for interactive environmental problems, such as mobility management and resource management, the MDP model is more suitable. Therefore, it is important to consider the advantages and disadvantages of each technique before applying them.

7.2. Implementation complexity of ML in KDN

In this subsection, we examine the implementation complexity of ML algorithms in KDN problems. Generally, four factors need to be considered while computing the complexity of an ML algorithm: (1) mathematical operations, (2) dataset, (3) data storage, and (4) software and hardware requirements. Mathematical operations vary from one problem statement to another and are usually based on the features of the ML algorithm that affect the complexity of the mathematical operation. Dataset complexity lies within the collection and use of appropriate data. Hence, collecting the necessary information from different wireless communication tasks has its own difficulties. For instance, in SL problems, data should be presented before finding a solution. In contrast, the data will be collected in RL as the agent interacts with the environment. Next, the complexity that occurs when the collected data or processed data are too large to be stored as knowledge in KP. Accordingly, software and hardware complexity arises from simulating the right software and using the correct number of processors or GPUs to process the knowledge, respectively.

7.3. Time consumption

One of the critical aspects of KDN is the response time. The ML algorithm tends to spend some time on training to provide near-optimal answers. Hence, it is essential to investigate the two-timing mechanisms that most ML techniques have, including training time and response time. The training time is the amount of time required for each ML algorithm to be fully trained. The response time is when an ML algorithm needs to make a prediction after being trained. Both timings are important for different applications in KDN networks [224].

The training time is most important for supervised and unsupervised learning, but it also has significance in RL for making accurate decisions. There are two different ways of training: online and offline. Depending on the application requirements, training may occur in one of the above processes. For example, highly dependent applications will perform online training, such as handover optimization, tuning, and healing in SONs. Applications incorporating NNs require time for adaptation to make accurate predictions, including mobility prediction and clustering problems [225]. Specifically, for applications with NN systems, the training time is usually longer [226]. Hence, the environment may change by the time the training algorithm learns a policy or mapping rule. In addition, training an RL model in complex wireless communication environments can be time-consuming and ineffective because the set of agents and environmental dynamics are different after the learning algorithm is completely trained. Therefore, such changes in the characteristics of the communication environment may affect the performance of the ML algorithm. However, the training time in NN-based approaches can be reduced using GPUs, and using TL helps to decrease the training time and accelerate the learning procedure, as illustrated in [227,228].

The response time of the trained learning algorithm is more important than the training time. Most applications in wireless networks require a quick response, on a timescale of milliseconds, such as decision making in resource management. Let us consider two different approaches of ML applied to these wireless network applications, namely NN-based approaches and alternative approaches. The time cost can be discussed as follows:

(1) *NN-based approaches:* In some applications, we can achieve an acceptable response time without GPU usage. For example, as shown in Table 1 in [89], on average, it is possible to achieve 0.0149 ms for power control decision making in a network with 30 users. Therefore, making resource management decisions within an acceptable time frame is feasible for power allocation problems using a trained DNN [89,229,230]. However, as the network size increased, both the response time and training time exponentially increased. One solution proposed by [231] is to use GPU-based parallel computing to enable NNs to predict within a tolerable range (milliseconds). Furthermore, there is a deep Q network in the DRL, and the deep Q network depends on the output of Q-values from the NN. Hence, the response time mostly depends on the NN process time. However, one promising solution that KDN naturally provides is using the pre-trained data as knowledge, so the response time can be near-optimal and suitable for future wireless network applications.

(2) *Alternative approaches:* In resource allocation problems, in both RL-based Q-learning and joint utility and strategy estimation-based approaches, the aim is to find a policy or strategy that suites the dynamic nature of the environment. In RL-based methods, after the learning algorithm converges, the policy or strategy from the trained algorithm becomes fixed. In Q-learning, the strategy is represented by a set of Q-values, where each set corresponds to a system state and an associated action. Therefore, a well-trained RL algorithm can respond in milliseconds. Additionally, the joint utility and strategy estimation-based learning goal is to choose a probability value that indicates an action. Here, the agent only needs to generate a random number between 0–1 to select the appropriate action. Consequently, a well-trained algorithm can accelerate the decision-making process and achieve a millisecond response.

7.4. Suitable dataset availability for training and generating knowledge

Different training data are collected or generated for supervised, unsupervised, and reinforcement learning, depending on the characteristics of the problem. For instance, in spectrum allocation problems, the authors of [82,83] used non-cooperative game methods where the data were trained based on the collected information from the primary and secondary users. In [86] the same non-cooperative spectrum allocation is modeled with the difference where the data is collected using an RNN model at each BS; here, BSs continuously interact with each other to collect training data. Other authors of [84] used joint utility and strategy estimation-based learning to collect D2D information and generate training data. In power allocation, when the SL method is used, such as in [89,229,230], the authors aim to adapt a neural network to approximate power allocation in a complex environment, including the genetic and WMMSE algorithms. Using these algorithms enables these studies to generate training data under different network scenarios that can later be used in KPs as knowledge. Obtaining appropriate raw data is important for researchers on the same topics. For example, in cache problems, the authors used different datasets for cache management to reduce the traffic caused by video streaming. In [110], the dataset used for training was acquired using the YouTube Application Programming Interface with 12500 YouTube videos. The dataset used to train the RNN in [111,112] was obtained from Youku [232] (a video hosting service based in Beijing, China) for content (video) request prediction and traffic management. Moreover, in [113], the dataset is a combination of two datasets, namely YUPENN [233] and UFC101 [233].

In networking scenarios, specifically in routing problems, the collected dataset is usually obtained from a simulator. For instance, in [137], the authors implemented a neural network using the Omnet++ simulator for traffic engineering. In this study, the SDN controller collects traffic flow reports from the nodes in the networks and separates different features to feed the SL algorithm. In [142], the authors investigated DNN-based routing to solve dynamic routing problems. The training dataset collects traffic patterns and paths that a packet will undertake to reach the destination in the proposed study. The routing paths were obtained using the traditional OSPF strategy from a software simulator. In other studies, such as in [141], the dataset was generated in real-time through online fashion with routers in the network for intelligent route selection.

In ML-based localization techniques, three main strategies are used to collect the required dataset and obtain the position of the object in an indoor environment: RSS, CSI, and ToA. Research studies in [209,210,215,217,218] used RSS information as a dataset, and studies in [219,220] used CSI data. The generated dataset was obtained using a cell phone or UWB device from real practical measurements. For example, the authors of [219] utilized the IWL 5300 NIC to mobile devices to read CSI data, while in [209], a client program was used in mobile devices to measure the RSS levels.

In RL, the learning algorithm continuously updates itself through interaction with the environment. In these problems, the learning agent takes action in a particular state. Then, it receives a reward from the environment, where the environment is created as a virtual environment by specific software, such as NS3, Matlab. For example, in spectrum management, researchers in [84] optimized the spectrum usage from reward feedback for each D2D pair. In power management, the authors of [105] used the reward as the difference between the maximum total power consumption and the current total power consumption to perform BS switching on/off to minimize the overall network power consumption.

7.5. Suggestions for choosing ML algorithms in KDN framework

As previously discussed, ML problems are generally categorized into regression, classification, clustering, and decision-making problems. In comparison, the differences between each lie within the solutions they provide for KDN problems. However, for each problem, there could be different ML solutions. Therefore, it is important to realize and compare ML algorithms, specifically those that can solve the same problem. This subsection provides a guideline for readers to first understand why recent studies have used ML algorithms (in the context of wireless networks) and provide guidance for selecting a suitable ML algorithm.

(1) *Techniques for deploy regression or classification:* For these two techniques, SVM, KNN, and NN are mostly used [111,112,140,193,194]. SVM is the most robust prediction method for binary classification with low complexity [113]. At the same time, KNN is a multiclass classifier, mostly known for its simplicity of implementation. In problems where the dataset is not linearly separable, KNN is a better classifier than SVM. In the KNN approach, only the distance metric and K parameter must be selected, whereas in SVM, the regularization and kernel parameters must be chosen carefully. Although these two algorithms are simple compared to neural networks, NN is more robust in feature extraction and improves overall network performance. For instance, a DNN can handle large datasets and achieve

high accuracy, although the learning procedure can be time-consuming owing to the optimization of various parameters. However, with sufficient training datasets and powerful GPUs, DNNs are more recommended than other learning machines. Furthermore, other neural networks, such as CNN and RNN, can reduce the training time and system overhead. Both algorithms have their own advantages for solving different problems. For instance, CNN is suitable for learning spatial features, including the channel gain matrix, while RNN is good at processing time series problems for feature extraction.

(2) *Techniques for deploy clustering:* ML models applied to clustering mainly include K-mean clustering. However, there are other supervised and NN studies deployed to clustering problems. K-mean clustering is one of the most popular and simplest methods for clustering data. Generally, the K-mean is used to differentiate between groups with similar data points and patterns. Neural networks are also used to organize the input data. For instance, given a set of images, the NN can organize and provide images with similar content. This process does not provide clusters, but it creates meaningful representations of the dataset, which can be used for clustering. The main difference between these two algorithms is the complexity of implementation. Moreover, in K-mean clustering, the number of cluster centroids (K value) is essential; however, in NN, the design of the hidden layers and other factors must be considered.

(3) *Techniques to deploy decision making:* ML-based decision-making algorithms are Q-learning, joint utility and strategy estimation, actor–critic, and DRL. These algorithms are applied to dynamic environments to learn patterns and decide accordingly. Q-learning is an off-policy RL that seeks to take the best action in any particular state. An actor–critic learning strategy is helpful in non-Markov environments, where the algorithm can learn an explicit stochastic policy. Compared with the Q-learning value-based learning algorithm, policy knowledge transfer is easier in actor–critic learning because policy and value functions are updated independently. Joint utility and strategy estimation-based learning are more suitable for multi-agent scenarios, providing a stable system when one agent diverges from its mixed strategy. The agent in DRL is capable of learning from a high-dimensional input state, which is the advantage of DRL over Q-learning and actor–critic learning [234–237]. Another advantage of DRL is its ability to make good action decisions even in unfamiliar situations [238]. Moreover, both Q-learning and actor–critic learning require a storage repository for each state and action procedure. Therefore, they are not suitable for multidimensional datasets. Furthermore, training a DRL has a high computational complexity.

8. Motivations for applying knowledge-defined networking

After summarizing the terms and conditions associated with ML algorithms, we need to investigate the motivations for applying the appropriate ML algorithm to KDN-based networks. It is essential to examine the reasons and motivations for adapting KDN-based approaches to wireless networks. The subsections below provide the reasons for applying ML and knowledge to the network based on the literature surveyed throughout this study.

8.1. Lack of network knowledge and intelligence

Although there exist centralized algorithms and techniques for optimization that can achieve the objectives of various performance criteria, the lack of global network knowledge and intelligence has been recognized by researchers [11,239]. For instance, the baseline technique to achieve load balancing and backhaul management in [120–122] requires complete information about the traffic load and content popularity of users before execution of cache content, which is challenging to acquire precise information in advance. Therefore, we suggest prior knowledge for decision-making. Moreover, using a TL algorithm, the past experience in cache content can be utilized by BSs to guide cache management even without knowing any information about the current traffic information. In data aggregation problems, supervised learning approaches can differentiate between data communication after a thorough training procedure. On the other hand, having knowledge about the network and the information that is being transmitted or received can help improve network performance and add intelligence [182]. Furthermore, in BS sleep-mode control problems, complete information of the network environment is required in advance, which is difficult to obtain. The authors of [228] used TL to adapt past information about BS switching to guide the current decision making for BS switching even without the knowledge of traffic loads in the network. In handover management, one solution is to use a fuzzy logic controller with a set of predefined rules, and each state of the system determines a specific action. However, the setting of each action mainly relies on the knowledge and information about the network environment, which might be unknown for the new state of the communication system. Overall, the main reason for adapting ML algorithms is to acquire knowledge from any particular system for optimization, self-organization, and self-healing. The majority of the papers surveyed in this study can be used as a prior processing technique to build a knowledge plane and provide intelligence for networks.

8.2. Self-organizing networks

Self-organizing networks (SONs) provide self-optimization, coordination, self-organization, and correction for the next generation of wireless networks [240]. In particular, most researchers now consider ML techniques as an official approach to achieve self-organization in the network, proving that KDN will be part of future wireless networks. 3GPP has already started developing protocols and technologies to automate network configurations [241]. In this context, RL is the most recognizable approach for correcting itself based on the environment and experience. In particular, in load balancing, handover management, routing, etc. In summary, SON can be applied using the following studies [81,85,86,95,99,106,107,114,133–135,184,192,196,200,205,242,243]. SON is an evolution of self-driving networks and KDN, and most applications in wireless networks require intelligence to tune, correct, and decide on behalf of human operators.

8.3. Achieving reliable and high performance compared to traditional optimization algorithms

Traditional optimization algorithms can only work for deterministic networks, which have certain characteristics. These algorithms are not practical in current real-life network scenarios, as the network traffic changes every day. In contrast, ML is the capability of machines to learn how to respond to any specific situation. Hence, they are more reliable than traditional algorithms owing to their flexibility and adaptation to new environments. In [80], DRL was utilized for spectrum allocation and was able to provide twice the channel throughput when compared to slotted-aloha with optimal probability. In [89], an SL algorithm was developed to train a DNN for power management, and it was shown to be superior to a state-of-the-art interference management algorithm. In cache management, the authors of [115] used the RL algorithm for content caching at the BS and compared it with two traditional cache update schemes, namely LRU and LFU, where it shows a better long-term cache hit rate. Moreover, in [173], the RL-based algorithm for user association performs much better than traditional dual-decomposition-based approaches. Overall, ML techniques have the potential to provide superior performance to traditional optimization algorithms. Other surveyed works for this motivation are as follows [92,109,115,121,169,173,219,230].

8.4. Learning patterns and predicting the future

By utilizing neural networks, hidden patterns in a system can be learned and used to estimate future values or predict the future, which is an advantage in KDN-based networks. In this context, the authors used this NN functionality to improve system performance. In [92], a multi-agent DRL technique was used to observe the spatial features based on the collected CSI and QoS to make a wiser power allocation decision when the network experiences dynamic changes. Moreover, in traditional indoor localization, the system's performance can be easily affected, resulting in inaccurate positioning owing to the complexity of the environment. As a result, many researchers are now motivated to use NNs to increase the positioning accuracy by learning and updating user patterns [214,219]. Consequently, the motivation of surveyed works utilizing NN can be observed in [12,86,89,92,97,111–113,116,117,141,164,168,193–195].

8.5. Ability to handle complex problems and provide low-complexity solutions

One of the important reasons and motivations for using ML algorithms is their ability to handle complex problems and datasets. The authors of [86] trained a multi-agent RL-based with ESN for efficient spectrum allocation and load balancing in LTE-U networks. Other researchers in [168] used CNN to achieve a low-complexity and high-accuracy clustering algorithm to classify three different types of waveforms in wireless communication systems. Moreover, the ability to handle high-complexity problems is the main reason why authors use RL. For example, for the on/off sleep mode control of small cells, the authors of [103] used distributed Q-learning to decide on the sleep mode of each BS, which led to a low-complexity sleep-mode control algorithm. Overall, the motivation for providing a low-complexity solution for the KDN paradigm can be seen in the literature [12,79,118,152,167,194,211,216,219].

8.6. Avoiding wrong decisions

Some traditional and heuristic approaches for optimization and estimation based on a fixed set of rules are often unable to avoid faulty and unsatisfactory results that have occurred previously. This means that these approaches are unable to learn from their mistakes and correct their decisions. Such problems can be seen in OSPF-based routing strategies, as stated in [141], where OSPF routing will result in some congestion in certain situations at some routers even though it may know the congested router. Hence, in these situations, the OSPF repeats the same action (wrong decision), leading to congestion again. In handover strategies based on RSSI values, a similar problem can accrue, as shown in [244]. Moreover, other similar problems can be observed in the literature for user association based on max-SINR [173] and in the BS on/off sleep mode control strategy [106]. To prevent wrong decision-making in traditional algorithms, RL and deep learning are adapted to learn through historical and new data to prevent any previously incorrect decisions. For instance, deep learning in routing strategies enables the algorithm to avoid mistakes, such as congestion in the network under different traffic patterns. RL can also be utilized in other approaches to overcome the same type of problem [245,246]. In summary, decision making with poor performance outcomes can be avoided using ML strategies in the KDN framework, which can be further observed in the surveyed literature of [63,89,95,98,114,115,118,133,134,141,150,247–251]

9. Challenges and open issues

There have been several studies on the applications of ML and its potential advantages for improving the overall network performance in wireless networks. However, few studies have focused on facilitating the ML-generated output data as knowledge, which can be used in similar applications and scenarios to create intelligence. Therefore, there remain many challenges and open issues that require attention from various sources across academia and industry to develop the concept of KDN standard criteria in future wireless networks. In this section, we identify the challenges and discuss future opportunities.

9.1. How the knowledge layer will be architecturally created

Several studies have introduced the KDN architecture, for instance, the authors of [28] restated the concept of KP in the context of SDN architecture in addition to the three planes of the SDN paradigm. Fig. 1 in this paper shows that the KP is located on top of the control and management planes. The integration of KP generates a behavioral model and reasoning process for decision-making. This architecture enables the KP to fully view and control the network via the control and management planes. Other research studies in [12–14,29] have a similar architecture of KDN. In [14], the same KP is utilized on top of all the layers, but it uses a cross-layer management and monitoring plane with ML algorithms to manage the rest of the planes. This paper [14] utilizes ML-based algorithms in both separate orchestration layers and embedded in the management plane. Therefore, there are different orchestrations of the KDN concept, and a thorough investigation is required to identify the most suitable one in terms of flexibility and performance.

9.2. Centralized, distributed, or hybrid KP

There are three available options for locating the knowledge plane in wireless networks: centralized, distributed, and hybrid. Identifying which KDN architecture is suitable for a problem is critical in implementing KDN in a real wireless network. In a centralized approach, devices only report the information to a centralized controller via protocols such as OF, P4, and network management. The controller processes this information using an ML technique to generate the knowledge. The controller uses two approaches to convey knowledge: direct and indirect approaches. In the direct approach, it uses the previously processed information (stored knowledge, not the typical network information) and immediately sends back the new strategies, while in the indirect approach, the ML algorithm obtains the new rules and then sends it to the devices. In the distributed approach, devices act by themselves based on the previously processed information or by communicating with each other to acquire the information required to create the knowledge. To obtain knowledge, specific ML-based algorithms are used for particular applications to optimize or predict. The obtained knowledge can be shared with other devices for better network decision-making. This knowledge can be exchanged between devices using D2D routing protocols. For instance, one particular ML can predict the best next-hop by utilizing cross-layer information between application and data layers. This prediction can be shared via multi-hop D2D communication with neighboring nodes to the rest of the network. Therefore, in multi-hop D2D networks, nodes will have prior knowledge about the best candidate nodes to relay their traffic to maximize overall network performance. The distributed KP might cause QoS and latency problems for packet delivery at the initial stage of the network. However, once nodes gather enough information about the other nodes, they can make near real-time decision-making that maximizes the QoS. In the hybrid approach, both the controller and devices act intelligently together based on the information collected. This information is used to acquire knowledge and inject a new rule based on the ML output.

9.3. Knowledge validation, uncertainty and compromises

ML and intelligence have been envisioned by many researchers as the most important feature in 6G, as ML algorithms have been extensively used in complex scenarios. Therefore, it is evident that the KDN architecture can be used to address the challenges of 6G. One of the main challenges faced by all technologies is the validation of knowledge. If 6G targets an automatically configured cellular network, there must be a mechanism to verify the confidence and certainty of knowledge. As a result, a certainty mechanism must be deployed to acknowledge the level of certainty, whether knowledge is practical or compromised. The output of the ML algorithm must be checked by the expected results to evaluate the degree of uncertainty. A threshold barrier can be used to validate the usefulness of knowledge. If the knowledge is authorized to deploy in the network, the ML output has been successful, but if the compared strategy has revealed unauthorized knowledge, then the ML's output cannot pass the threshold value. In this case, a new ML technique must compute the new knowledge and go through the same procedure, which causes a delay that affects the system performance. In the worst-case scenario, if the knowledge is rejected again, then an extreme case must be considered. To mitigate the worst-case scenario, any proposed algorithm must undergo various experimental analyses in a real testbed to find the possible ML substitutes in any given scenario. Therefore, ML algorithms must be tested in the same environment with similar characteristics to provide insights into different ML techniques. Consequently, those with similar performance will be selected to be each other's replacement to mitigate the worst-case scenario. Moreover, related regulations and standardization must be established to guarantee the 6G service requirements.

9.4. Lapse, or loss, or priority of knowledge

There are two approaches to ML: reactive and proactive. In the reactive approach, the learning agent must address an issue that has already been defined. However, in a proactive manner, the learning agent must simultaneously adapt to possible future problems. Current network functions are mainly designed using reactive network protocols with human intervention for maintenance and upgrading. To enable intelligence and deploy KDN in future wireless networks, proactive learning ability is necessary [252]. In proactive learning, knowledge must be prioritized such that important metrics of the network, including channel allocation, traffic clustering, traffic prediction, computing offloading, radio resource scheduling, and network configuration, have a ranking of knowledge prioritization. Thus, once the network is fully congested with heterogeneous nodes, the KP must prioritize which tasks have the most pressing matter and which ones are crucial for the network functionality. Another reason for prioritization could be the lack of storage for knowledge, where it can cause lapse or, in a worst-case scenario, loss of knowledge. A lapse of knowledge is referred to as incomplete knowledge throughout prioritization, ML training, or technical issues. Moreover, loss of knowledge can accrue due to memory shortages and technical issues. In both cases, knowledge is useless and has a negative impact on the network performance. Therefore, a systematic monitoring management module must observe the KP plane. Specifically, to ensure that each network application is prioritized for knowledge extraction, the storage capacity is sufficient, and the knowledge is stable and informative.

9.5. Effectiveness of an ML algorithm in KDN

Although ML studies have been a point of discussion over the last few years [253], current research in the wireless communication area is still unripe. This is the main reason why ML has not been practically applied to existing wireless networks. To evolve ML-based algorithms to meet the requirements of future 6G systems, it is essential to standardize AI-embedded communication. The performance of the current ML-based algorithms is evaluated based on the level of improvement in communication performance. However, to adapt KDN and intelligence across 6G networks, it is far from sufficient to only consider the degree of improvement without considering computation and storage costs. Therefore, in addition to evaluating the performance of ML technologies, the required storage and computation overhead must be considered while designing the standards. Moreover, to realize full intelligence in KDN networks, the compatibility of developed ML algorithms with other network functions is another emerging topic. Further, the intelligence and ML algorithms must be adaptive to any changes in the topology to enable automatic adjustment in 6G networks [254].

9.6. Standard baseline dataset for experiments

In wireless communication problems, one of the fundamental aspects of an effective ML-based algorithm is the representation of a dataset. The training procedure is the step in which the effectiveness of the ML algorithm is determined. In this step, the dataset provided to the ML algorithm has a direct impact on the usefulness of ML. In wireless networks, terabytes of information can be collected; however, if the machine cannot make sense of the recorded data, the ML approach is useless. Therefore, data-related issues are one of the main problems that some ML-based projects cannot be accomplished [255]. In the context of ML research projects, some concluded that no relevant dataset was available. Therefore, the process is either highly time-consuming or challenging to analyze. In self-driving vehicles, SL has shown promising results in the future of the vehicle industry [256]. However, the performance of these algorithms relies on the size of the dataset to provide a reliable answer. Moreover, in this type of problem, the response time of the algorithm is very important, which makes the algorithm more complicated. This type of issue occurs across different wireless communication problems. Hence, creating a standard dataset and platform for researchers is very important for generating a reliable and effective solution for 6G and beyond. This will enable researchers interested in one topic area to work on the same dataset with the same platforms. Moreover, ML is used to extract features and learn the rules from large datasets automatically, introducing another challenge that needs to be addressed by how much manual intervention and repeated work need to be reduced to have a fully functional ML algorithm. Therefore, it is crucial for future wireless communication systems to have a reliable dataset and platform for researchers to work on ML-based wireless network problems with certainty.

10. Conclusion

One of the crucial aspects in 6G wireless networks is intelligence, and many research studies are now focusing on exploring how knowledge and intelligence can be integrated into wireless networks. This survey paper investigated the concept of knowledge-defined networking, which aims to combine SDN and ML/AI to create a programmable and knowledge-aware networking architecture. We first introduced emerging technologies to facilitate KDN, specifically the SDN paradigm, network telemetry, and ML algorithms. We then investigated most of the widespread applications of wireless networks. The reviewed studies in network applications were based on the most recent ML-based approached to create automated applications in KDN-based wireless networks. The applications were categorized into the MAC layer, Network layer, and Application layer. Resource management problems were distributed within the MAC layer and classified as spectrum allocation, power management, QoS, BS switching, cache, and backhaul management. Networking and mobility management problems were investigated in the network layer. Networking problems were described as routing strategies, clustering, user/BS association, traffic classification, and data aggregation. Mobility prediction and handover management were considered in mobility management. Then, from the Application layer perspective, various indoor localization techniques are presented. Moreover, appropriate ML-based studies were thoroughly explored for each surveyed application, and the most suitable KDN architecture was suggested. We achieved a comprehensive review of different parts of wireless networks and provided insights into how different algorithms perform, enabling future researchers to adapt the most appropriate ML-based study with the suitable architecture of KDN. Further, the conditions associated with ML-based strategies in the context of KDN were provided, followed by the motivation to apply KDN. Finally, we outlined several unsolved problems and challenges within the KDN paradigm.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Shafin R, Liu L, Chandrasekhar V, Chen H, Reed J, Zhang JC. Artificial intelligence-enabled cellular networks: A critical path to beyond-5G and 6G. IEEE Wirel Commun 2020;27(2):212–7.
[2] Jiang W, Han B, Habibi MA, Schotten HD. The road towards 6G: A comprehensive survey. IEEE Open J Commun Soc 2021;2:334–66.
[3] Clark DD, Partridge C, Ramming JC, Wroclawski JT. A knowledge plane for the internet. In: Proceedings of the 2003 conference on applications, technologies, architectures, and protocols for computer communications. ACM; 2003, p. 3–10.
[4] Mestres A, Rodriguez-Natal A, Carner J, Barlet-Ros P, Alarcón E, Solé M, et al. Knowledge-defined networking. ACM SIGCOMM Comput Commun Rev 2017;47(3):2–10.
[5] Bosshart P, Daly D, Gibb G, Izzard M, McKeown N, Rexford J, et al. P4: Programming protocol-independent packet processors. ACM SIGCOMM Comput Commun Rev 2014;44(3):87–95.
[6] Hu F, Hao Q, Bao K. A survey on software-defined network and openflow: From concept to implementation. IEEE Commun Surv Tutor 2014;16(4):2181–206.
[7] Niknam S, Roy A, Dhillon HS, Singh S, Banerji R, Reed JH, et al. Intelligent o-ran for beyond 5g and 6g wireless networks. 2020, arXiv preprint arXiv:2005.08374.
[8] Pan C, Elkashlan M, Wang J, Yuan J, Hanzo L. User-centric C-RAN architecture for ultra-dense 5G networks: Challenges and methodologies. IEEE Commun Mag 2018;56(6):14–20.
[9] Kazemifard N, Shah-Mansouri V. Minimum delay function placement and resource allocation for open RAN (O-RAN) 5G networks. Comput Netw 2021;188:107809.
[10] Balasubramanian B, Daniels ES, Hiltunen M, Jana R, Joshi K, Sivaraj R, et al. RIC: A RAN intelligent controller platform for AI-enabled cellular networks. IEEE Internet Comput 2021;25(2):7–17.
[11] Kato N, Mao B, Tang F, Kawamoto Y, Liu J. Ten challenges in advancing machine learning technologies toward 6G. IEEE Wirel Commun 2020;27(3):96–103.
[12] Pham TAQ, Hadjadj-Aoul Y, Outtagarts A. Deep reinforcement learning based QoS-aware routing in knowledge-defined networking. In: International conference on heterogeneous networking for quality, reliability, security and robustness. Springer; 2018, p. 14–26.
[13] Lu W, Liang L, Kong B, Li B, Zhu Z. AI-assisted knowledge-defined network orchestration for energy-efficient data center networks. IEEE Commun Mag 2020;58(1):86–92.
[14] Careglio D, Spadaro S, Cabellos A, Lazaro J, Perelló J, Barlet P, et al. ALLIANCE project: Architecting a knowledge-defined 5G-enabled network infrastructure. In: 2018 20th International conference on transparent optical networks. IEEE; 2018, p. 1–6.
[15] Hyun J, Van Tu N, Hong JW-K. Towards knowledge-defined networking using in-band network telemetry. In: NOMS 2018-2018 IEEE/IFIP network operations and management symposium. IEEE; 2018, p. 1–7.
[16] Liu G, Jiang D. 5G: Vision and requirements for mobile communication system towards year 2020. Chin J Eng 2016;2016(2016):8.
[17] Bhalla MR, Bhalla AV. Generations of mobile wireless technology: A survey. Int J Comput Appl 2010;5(4):26–32.
[18] Marshall C. By 2019, 80% of the world's internet traffic will be video. Tubular Insights 2015;3:2017, Retrieved on January.
[19] Cass S. The age of the zettabyte cisco: the future of internet traffic is video [Dataflow]. IEEE Spectr 2014;51(3):68.
[20] Abrol A, Jha RK, Jain S, Kumar P. Joint power allocation and relay selection strategy for 5G network: a step towards green communication. Telecommun Syst 2018;68(2):201–15.
[21] Ning Z, Wang X, Rodrigues JJ, Xia F. Joint computation offloading, power allocation, and channel assignment for 5G-enabled traffic management systems. IEEE Trans Ind Inf 2019;15(5):3058–67.
[22] Rana DS, Dhondiyal SA, Chamoli SK. Software defined networking (SDN) challenges, issues and solution. Int J Comput Sci Eng 2019;7(1):884–9.
[23] Rowshanrad S, Namvarasl S, Abdi V, Hajizadeh M, Keshtgary M. A survey on SDN, the future of networking. J Adv Comput Sci Technol 2014;3(2):232–48.
[24] Haque IT, Abu-Ghazaleh N. Wireless software defined networking: A survey and taxonomy. IEEE Commun Surv Tutor 2016;18(4):2713–37.
[25] Kim C, Sivaraman A, Katta N, Bas A, Dixit A, Wobker LJ. In-band network telemetry via programmable dataplanes. In: ACM SIGCOMM, Vol. 15. 2015.
[26] Sun Y, Peng M, Zhou Y, Huang Y, Mao S. Application of machine learning in wireless networks: Key techniques and open issues. 2018, arXiv preprint arXiv:1809.08707.
[27] Mao H, Alizadeh M, Menache I, Kandula S. Resource management with deep reinforcement learning. In: Proceedings of the 15th ACM workshop on hot topics in networks. ACM; 2016, p. 50–6.
[28] Hyun J, Hong JW-K. Knowledge-defined networking using in-band network telemetry. In: 2017 19th Asia-Pacific network operations and management symposium. IEEE; 2017, p. 54–7.
[29] Duque-Torres A, Amezquita-Suárez F, Caicedo Rendon OM, Ordóñez A, Campo WY. An approach based on knowledge-defined networking for identifying heavy-hitter flows in data center networks. Appl Sci 2019;9(22):4808.
[30] Benson T, Akella A, Maltz DA. Unraveling the complexity of network management. In: NSDI. 2009, p. 335–48.
[31] Kobo HI, Abu-Mahfouz AM, Hancke GP. A survey on software-defined wireless sensor networks: Challenges and design requirements. IEEE Access 2017;5:1872–99.
[32] Galán-Jiménez J, Gazo-Cervero A. Overlay networks: overview, applications and challenges. IJCSNS 2010;10(12):40.
[33] Tarkoma S. Overlay networks: Toward information networking. Auerbach Publications; 2010.
[34] Kreutz D, Ramos F, Verissimo P, Rothenberg CE, Azodolmolky S, Uhlig S. Software-defined networking: A comprehensive survey. 2014, arXiv preprint arXiv:1406.0440.
[35] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, et al. Openflow: enabling innovation in campus networks. ACM SIGCOMM Comput Commun Rev 2008;38(2):69–74.
[36] Goransson P, Black C, Culver T. Software defined networks: A comprehensive approach. Morgan Kaufmann; 2016.
[37] Lara A, Kolasani A, Ramamurthy B. Network innovation using openflow: A survey. IEEE Commun Surv Tutor 2013;16(1):493–512.
[38] Tourrilhes J, Sharma P, Banerjee S, Pettit J. Sdn and openflow evolution: A standards perspective. Computer 2014;47(11):22–9.
[39] Patterson L. SDN: The readiness of open source frameworks for production networking [PhD Thesis], 2018.
[40] Nick M, Jen R. Clarifying the differences between P4 and OpenFlow. 2016, [Online]. Available: https://p4.org/p4/clarifying-the-differences-between-p4-and-openflow.html.
[41] Stubbe H. P4 compiler & interpreter: A survey. In: Proc. future internet (FI) innov. internet technol. mobile commun., Vol. 47. 2017, p. 1–72.
[42] Well MBH, Optimizations B. P4 data plane programming for server-based networking applications.
[43] Signorello S, State R, François J, Festor O. Ndn. p4: Programming information-centric data-planes. In: 2016 IEEE netsoft conference and workshops. IEEE; 2016, p. 384–9.
[44] Antonin B. Behavioral model (bmv2). 2016, [Online]. Available: https://github.com/p4lang/behavioral-model.
[45] Laki S, Horpácsi D, Vörös P, Kitlei R, Leskó D, Tejfel M. High speed packet forwarding compiled from protocol independent data plane specifications. In: Proceedings of the 2016 ACM SIGCOMM conference. ACM; 2016, p. 629–30.
[46] Shahbaz M, Choi S, Pfaff B, Kim C, Feamster N, McKeown N, Rexford J. Pisces: A programmable, protocol-independent software switch. In: Proceedings of the 2016 ACM SIGCOMM conference. ACM; 2016, p. 525–38.
[47] Kheradmand A, Rosu G. P4K: A formal semantics of P4 and applications. 2018, arXiv preprint arXiv:1804.01468.
[48] Sultana N, Chilamkurti N, Peng W, Alhadad R. Survey on SDN based network intrusion detection system using machine learning approaches. Peer-To-Peer Netw Appl 2019;12(2):493–501.
[49] Michalski R, Carbonell J, Mitchell T. Machine learning: An artificial intelligence approach, springer science & business media. 2013.
[50] Kotsiantis SB, Zaharakis I, Pintelas P. Supervised machine learning: A review of classification techniques. Emerg Artif Intell Appl Comput Eng 2007;160(1):3–24.
[51] Criminisi A, Shotton J, Konukoglu E, et al. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. 2012.
[52] Molnar C. Interpretable machine learning. Lulu. com; 2019.
[53] Scholkopf B, Smola AJ. Learning with kernels: Support vector machines, regularization, optimization, and beyond. MIT Press; 2001.

[54] Vapnik VN. An overview of statistical learning theory. IEEE Trans Neural Netw 1999;10(5):988–99.
[55] Mohammed M, Khan MB, Bashier EBM. Machine learning: algorithms and applications. Crc Press; 2016.
[56] Celebi ME, Aydin K. Unsupervised learning algorithms. Springer; 2016.
[57] Kim K. Face recognition using principle component analysis. In: International conference on computer vision and pattern recognition, Vol. 586. 1996, p. 591.
[58] Madsen RE, Hansen LK, Winther O. Singular value decomposition and principal component analysis. Neural Netw 2004;1:1–5.
[59] Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT Press; 2018.
[60] Sutton RS, Barto AG, et al. Introduction to reinforcement learning, Vol. 2, no. 4. MIT press Cambridge; 1998.
[61] Glorennec PY. Fuzzy Q-learning and dynamical fuzzy Q-learning. In: Proceedings of 1994 IEEE 3rd international fuzzy systems conference. IEEE; 1994, p. 474–9.
[62] Goodfellow I, Bengio Y, Courville A. Deep learning, book in preparation for MIT press. 2016, [Online]. Available: http://www.deeplearningbook.org/.
[63] Ye H, Li GY. Deep reinforcement learning based distributed resource allocation for V2V broadcasting. In: 2018 14th International wireless communications & mobile computing conference. IEEE; 2018, p. 440–5.
[64] Ye H, Li GY, Juang B-H. Power of deep learning for channel estimation and signal detection in OFDM systems. IEEE Wireless Commun Lett 2017;7(1):114–7.
[65] Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, et al. Playing atari with deep reinforcement learning. 2013, arXiv preprint arXiv:1312.5602.
[66] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, et al. Mastering the game of go with deep neural networks and tree search. Nature 2016;529(7587):484.
[67] Li Y. Deep reinforcement learning: An overview. 2017, arXiv preprint arXiv:1701.07274.
[68] Karagiannakos S. The idea behind actor-critics and how A2C and A3C improve them. 2018, [Online]. Available: https://sergioskar.github.io/Actor-critics/.
[69] Singh S, Jaakkola T, Littman ML, Szepesvári C. Convergence results for single-step on-policy reinforcement-learning algorithms. Mach Learn 2000;38(3):287–308.
[70] Alpcan T, Boche H, Honig ML, Poor HV. Mechanisms and games for dynamic spectrum allocation. Cambridge University Press; 2013.
[71] Perlaza SM, Tembine H, Lasaulce S. How can ignorant but patient cognitive terminals learn their strategy and utility? In: 2010 IEEE 11th international workshop on signal processing advances in wireless communications. IEEE; 2010, p. 1–5.
[72] Anthony M, Bartlett PL. Neural network learning: Theoretical foundations. cambridge University Press; 2009.
[73] Ruder S. An overview of gradient descent optimization algorithms. 2016, arXiv preprint arXiv:1609.04747.
[74] Niepert M, Ahmed M, Kutzkov K. Learning convolutional neural networks for graphs. In: International conference on machine learning. 2016, p. 2014–23.
[75] Yamashita R, Nishio M, Do RKG, Togashi K. Convolutional neural networks: an overview and application in radiology. Insights Imaging 2018;9(4):611–29.
[76] Islam M, Chen G, Jin S. An overview of neural network. Am J Neural Netw Appl 2019;5(1):7–11.
[77] Taylor ME, Stone P. Transfer learning for reinforcement learning domains: A survey. J Mach Learn Res 2009;10(Jul):1633–85.
[78] Kazmi SA, Tran NH, Saad W, Le LB, Ho TM, Hong CS. Optimized resource management in heterogeneous wireless networks. IEEE Commun Lett 2016;20(7):1397–400.
[79] Kaur A, Kumar K. A reinforcement learning based evolutionary multi-objective optimization algorithm for spectrum allocation in cognitive radio networks. Phys Commun 2020;43:101196.
[80] Naparstek O, Cohen K. Deep multi-user reinforcement learning for distributed dynamic spectrum access. IEEE Trans Wireless Commun 2018;18(1):310–23.
[81] Srinivasan M, Kotagi VJ, Murthy CSR. A Q-learning framework for user QoE enhanced self-organizing spectrally efficient network using a novel inter-operator proximal spectrum sharing. IEEE J Sel Areas Commun 2016;34(11):2887–901.
[82] Fan C, Li B, Zhao C, Guo W, Liang Y-C. Learning-based spectrum sharing and spatial reuse in mm-wave ultradense networks. IEEE Trans Veh Technol 2017;67(6):4954–68.
[83] Zhang Y, Tay WP, Li KH, Esseghir M, Gaïti D. Learning temporal–spatial spectrum reuse. IEEE Trans Commun 2016;64(7):3092–103.
[84] Sun Y, Peng M, Poor HV. A distributed approach to improving spectral efficiency in uplink device-to-device-enabled cloud radio access networks. IEEE Trans Commun 2018;66(12):6511–26.
[85] Alnwaimi G, Vahid S, Moessner K. Dynamic heterogeneous learning games for opportunistic access in LTE-based macro/femtocell deployments. IEEE Trans Wireless Commun 2014;14(4):2294–308.
[86] Chen M, Saad W, Yin C. Echo state networks for self-organizing resource allocation in LTE-U with uplink–downlink decoupling. IEEE Trans Wireless Commun 2016;16(1):3–16.
[87] Lukoševičius M. A practical guide to applying echo state networks. In: Neural networks: Tricks of the trade. Springer; 2012, p. 659–86.
[88] Chatzidimitriou KC, Partalas I, Mitkas PA, Vlahavas I. Transferring evolved reservoir features in reinforcement learning tasks. In: European workshop on reinforcement learning. Springer; 2011, p. 213–24.
[89] Sun H, Chen X, Shi Q, Hong M, Fu X, Sidiropoulos ND. Learning to optimize: Training deep neural networks for interference management. IEEE Trans Signal Process 2018;66(20):5438–53.
[90] Shi Q, Razaviyayn M, Luo Z-Q, He C. An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel. IEEE Trans Signal Process 2011;59(9):4331–40.
[91] Baligh H, Hong M, Liao W-C, Luo Z-Q, Razaviyayn M, Sanjabi M, et al. Cross-layer provision of future cellular networks: A WMMSE-based approach. IEEE Signal Process Mag 2014;31(6):56–68.
[92] Nasir YS, Guo D. Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks. IEEE J Sel Areas Commun 2019;37(10):2239–50.
[93] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature 2015;518(7540):529–33.
[94] Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: A survey. J Artificial Intelligence Res 1996;4:237–85.
[95] Simsek M, Bennis M, Güvenç I. Learning based frequency-and time-domain inter-cell interference coordination in HetNets. IEEE Trans Veh Technol 2014;64(10):4589–602.
[96] Asheralieva A, Miyanaga Y. An autonomous learning-based algorithm for joint channel and power level selection by D2D pairs in heterogeneous cellular networks. IEEE Trans Commun 2016;64(9):3996–4012.
[97] Zhang H, Yang N, Huangfu W, Long K, Leung VC. Power control based on deep reinforcement learning for spectrum sharing. IEEE Trans Wireless Commun 2020.
[98] Sanguanpuak T, Guruacharya S, Rajatheva N, Bennis M, Latva-Aho M. Multi-operator spectrum sharing for small cell networks: A matching game perspective. IEEE Trans Wireless Commun 2017;16(6):3761–74.
[99] Bennis M, Perlaza SM, Blasco P, Han Z, Poor HV. Self-organization in small cell networks: A reinforcement learning approach. IEEE Trans Wireless Commun 2013;12(7):3202–12.
[100] Zhu G, Zan J, Yang Y, Qi X. A supervised learning based QoS assurance architecture for 5G networks. IEEE Access 2019;7:43598–606.
[101] Wu H, Zhang Z, Luo J, Yue K, Hsu C-H. Multiple attributes QoS prediction via deep neural model with contexts. IEEE Trans Serv Comput 2018.
[102] Feng M, Mao S, Jiang T. Base station ON-OFF switching in 5G wireless networks: Approaches and challenges. IEEE Wirel Commun 2017;24(4):46–54.
[103] Gan X, Wang L, Feng X, Liu J, Yu H, Zhang Z, et al. Energy efficient switch policy for small cells. China Commun 2015;12(1):78–88.
[104] Li R, Zhao Z, Chen X, Zhang H. Energy saving through a learning framework in greener cellular radio access networks. In: 2012 IEEE global communications conference. IEEE; 2012, p. 1556–61.
[105] Xu Z, Wang Y, Tang J, Wang J, Gursoy MC. A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs. In: 2017 IEEE international conference on communications. IEEE; 2017, p. 1–6.
[106] Yu G, Chen Q, Yin R. Dual-threshold sleep mode control scheme for small cells. IET Commun 2014;8(11):2008–16.
[107] Fan S, Tian H, Sengul C. Self-optimized heterogeneous networks for energy efficiency. EURASIP J Wireless Commun Networking 2015;2015(1):21.
[108] Index CVN. Global mobile data traffic forecast update, 2015–2020. 2016, p. 9, Cisco White Paper.
[109] Chang Z, Lei L, Zhou Z, Mao S, Ristaniemi T. Learn to cache: Machine learning for network edge caching in the big data era. IEEE Wirel Commun 2018;25(3):28–35.
[110] Tanzil SS, Hoiles W, Krishnamurthy V. Adaptive scheme for caching YouTube content in a cellular network: Machine learning approach. IEEE Access 2017;5:5870–81.
[111] Chen M, Mozaffari M, Saad W, Yin C, Debbah M, Hong CS. Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience. IEEE J Sel Areas Commun 2017;35(5):1046–61.
[112] Chen M, Saad W, Yin C, Debbah M. Echo state networks for proactive caching in cloud-based radio access networks with mobile users. IEEE Trans Wireless Commun 2017;16(6):3520–35.
[113] Doan KN, Van Nguyen T, Quek TQ, Shin H. Content-aware proactive caching for backhaul offloading in cellular network. IEEE Trans Wireless Commun 2018;17(5):3128–40.
[114] Tamoor-ul Hassan S, Samarakoon S, Bennis M, Latva-Aho M, Hong CS. Learning-based caching in cloud-aided wireless networks. IEEE Commun Lett 2017;22(1):137–40.
[115] Wang W, Lan R, Gu J, Huang A, Shan H, Zhang Z. Edge caching at base stations with device-to-device offloading. IEEE Access 2017;5:6399–410.
[116] Li L, Xu Y, Yin J, Liang W, Li X, Chen W, et al. Deep reinforcement learning approaches for content caching in cache-enabled D2D networks. IEEE Internet Things J 2019;7(1):544–57.

[117] He Y, Zhang Z, Yu FR, Zhao N, Yin H, Leung VC, et al. Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks. IEEE Trans Veh Technol 2017;66(11):10433–45.
[118] Zhong C, Gursoy MC, Velipasalar S. A deep reinforcement learning-based framework for content caching. In: 2018 52nd Annual conference on information sciences and systems. IEEE; 2018, p. 1–6.
[119] He Y, Zhao N, Yin H. Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach. IEEE Trans Veh Technol 2017;67(1):44–55.
[120] Baştuğ E, Bennis M, Debbah M. A transfer learning approach for cache-enabled wireless networks. In: 2015 13th International symposium on modeling and optimization in mobile, Ad Hoc, and wireless networks. IEEE; 2015, p. 161–6.
[121] Bastug E, Bennis M, Debbah M. Anticipatory caching in small cell networks: A transfer learning approach. In: 1st KuVS workshop on anticipatory networks. 2014.
[122] Bharath B, Nagananda KG, Poor HV. A learning-based approach to caching in heterogenous small cell networks. IEEE Trans Commun 2016;64(4):1674–86.
[123] Venables WN, Ripley BD. Modern applied statistics with S-PLUS. Springer Science & Business Media; 2013.
[124] Hyvärinen A, Karhunen J, Oja E. Independent component analysis, vol. 46. Hoboken. NJ, USA: Wiley; 2004.
[125] Abrams M, Standridge CR, Abdulla G, Williams S, Fox EA. Caching proxies: Limitations and potentials. 1995.
[126] Arlitt M, Cherkasova L, Dilley J, Friedrich R, Jin T. Evaluating content management techniques for web proxy caches. ACM SIGMETRICS Perform Eval Rev 2000;27(4):3–11.
[127] Psounis K, Prabhakar B. A randomized web-cache replacement scheme. In: Proceedings IEEE INFOCOM 2001. Conference on computer communications. Twentieth annual joint conference of the IEEE computer and communications society (Cat. No. 01CH37213), Vol. 3. IEEE; 2001, p. 1407–15.
[128] Dulac-Arnold G, Evans R, van Hasselt H, Sunehag P, Lillicrap T, Hunt J, et al. Deep reinforcement learning in large discrete action spaces. 2015, arXiv preprint arXiv:1512.07679.
[129] Lee D, Choi J, Kim J-H, Noh SH, Min SL, Cho Y, et al. On the existence of a spectrum of policies that subsumes the least recently used (LRU) and least frequently used (LFU) policies. In: Proceedings of the 1999 ACM SIGMETRICS international conference on measurement and modeling of computer systems. 1999, p. 134–43.
[130] Bastug E, Bennis M, Debbah M. Living on the edge: The role of proactive caching in 5G wireless networks. IEEE Commun Mag 2014;52(8):82–9.
[131] Blaszczyszyn B, Giovanidis A. Optimal geographic caching in cellular networks. In: 2015 IEEE international conference on communications. IEEE; 2015, p. 3358–63.
[132] Liu A, Lau VK. Exploiting base station caching in MIMO cellular networks: Opportunistic cooperation for video streaming. IEEE Trans Signal Process 2014;63(1):57–69.
[133] Hamidouche K, Saad W, Debbah M, Song JB, Hong CS. The 5G cellular backhaul management dilemma: To cache or to serve. IEEE Trans Wireless Commun 2017;16(8):4866–79.
[134] Jaber M, Imran M, Tafazolli R, Tukmanov A. An adaptive backhaul-aware cell range extension approach. In: 2015 IEEE international conference on communication workshop. IEEE; 2015, p. 74–9.
[135] Xu Y, Yin R, Yu G. Adaptive biasing scheme for load balancing in backhaul constrained small cell networks. IET Commun 2015;9(7):999–1005.
[136] Fadlullah ZM, Tang F, Mao B, Kato N, Akashi O, Inoue T, et al. State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. IEEE Commun Surv Tutor 2017;19(4):2432–55.
[137] Yanjun L, Xiaobo L, Osamu Y. Traffic engineering framework with machine learning based meta-layer in software-defined networks. In: 2014 4th IEEE international conference on network infrastructure and digital content. IEEE; 2014, p. 121–5.
[138] Chen Y-S, Hsu C-S, Siao Y-G. Linear regression-based delay-bounded routing protocols for VANETs. Wireless Commun Mob Comput 2014;14(2):186–99.
[139] Wang Y, Martonosi M, Peh L-S. A supervised learning approach for routing optimizations in wireless sensor networks. In: Proceedings of the 2nd international workshop on multi-hop ad hoc networks: From theory to reality. 2006, p. 79–86.
[140] Sharma DK, Dhurandher SK, Woungang I, Srivastava RK, Mohananey A, Rodrigues JJ. A machine learning-based protocol for efficient routing in opportunistic networks. IEEE Syst J 2016;12(3):2207–13.
[141] Tang F, Mao B, Fadlullah ZM, Kato N, Akashi O, Inoue T, et al. On removing routing protocol from future wireless networks: A real-time deep learning approach for intelligent traffic control. IEEE Wirel Commun 2017;25(1):154–60.
[142] Kato N, Fadlullah ZM, Mao B, Tang F, Akashi O, Inoue T, et al. The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective. IEEE Wirel Commun 2016;24(3):146–53.
[143] Yao H, Yuan X, Zhang P, Wang J, Jiang C, Guizani M. A machine learning approach of load balance routing to support next-generation wireless networks. In: 2019 15th International wireless communications & mobile computing conference. IEEE; 2019, p. 1317–22.
[144] Sharma DK, Dhurandher SK, Agarwal D, Arora K. KROp: k-means clustering based routing protocol for opportunistic networks. J Ambient Intell Humaniz Comput 2019;10(4):1289–306.
[145] Tang Y, Cheng N, Wu W, Wang M, Dai Y, Shen X. Delay-minimization routing for heterogeneous VANETs with machine learning based mobility prediction. IEEE Trans Veh Technol 2019;68(4):3967–79.
[146] Zhao L, Wang J, Liu J, Kato N. Routing for crowd management in smart cities: A deep reinforcement learning perspective. IEEE Commun Mag 2019;57(4):88–93.
[147] Kadam K, Srivastava N. Application of machine learning (reinforcement learning) for routing in wireless sensor networks (WSNs). In: 2012 1st International symposium on physics and technology of sensors. IEEE; 2012, p. 349–52.
[148] Saleem Y, Yau K-LA, Mohamad H, Ramli N, Rehmani MH, Ni Q. Clustering and reinforcement-learning-based routing for cognitive radio networks. IEEE Wirel Commun 2017;24(4):146–51.
[149] Syed AR, Yau K-LA, Qadir J, Mohamad H, Ramli N, Keoh SL. Route selection for multi-hop cognitive radio networks using reinforcement learning: An experimental study. IEEE Access 2016;4:6304–24.
[150] Al-Rawi HA, Yau K-LA, Mohamad H, Ramli N, Hashim W. Effects of network characteristics on learning mechanism for routing in cognitive radio ad hoc networks. In: 2014 9th International symposium on communication systems, networks & digital sign. IEEE; 2014, p. 748–53.
[151] Tilwari V, Dimyati K, Hindia M, Fattouh A, Amiri IS. Mobility, residual energy, and link quality aware multipath routing in MANETs with Q-learning algorithm. Appl Sci 2019;9(8):1582.
[152] Yao H, Tianle M, Chunxiao J, Linling K, Song G. AI routers & network mind: A hybrid machine learning paradigm for packet routing. IEEE Comput Intell Mag 2019;14(4):21–30.
[153] Stampa G, Arias M, Sánchez-Charles D, Muntés-Mulero V, Cabellos A. A deep-reinforcement learning approach for software-defined networking routing optimization. 2017, arXiv preprint arXiv:1709.07080.
[154] Stützle T, Hoos HH. MAX–MIN ant system. Future Gener Comput Syst 2000;16(8):889–914.
[155] Sim KM, Sun WH. Ant colony optimization for routing and load-balancing: survey and new directions. IEEE Trans Syst, Man, Cybern A: Syst Hum 2003;33(5):560–72.
[156] Dhurandher SK, Sharma DK, Woungang I, Bhati S. HBPR: history based prediction for routing in infrastructure-less opportunistic networks. In: 2013 IEEE 27th international conference on advanced information networking and applications. IEEE; 2013, p. 931–6.
[157] Lindgren A, Doria A, Schelén O. Probabilistic routing in intermittently connected networks. ACM SIGMOBILE Mob Comput Commun Rev 2003;7(3):19–20.
[158] Boyan JA, Littman ML. Packet routing in dynamically changing networks: A reinforcement learning approach. In: Advances in neural information processing systems. 1994, p. 671–8.
[159] Yi J, Adnane A, David S, Parrein B. Multipath optimized link state routing for mobile ad hoc networks. Ad Hoc Netw 2011;9(1):28–47.
[160] Yi J, Parrein B. Multipath extension for the optimized link state routing protocol version 2 (OLSRv2). 2017.
[161] Guo H, Wei T. Logistic regression for imbalanced learning based on clustering. Int J Comput Sci Eng 2019;18(1):54–64.
[162] Subudhi S, Panigrahi S. Use of fuzzy clustering and support vector machine for detecting fraud in mobile telecommunication networks. Int J Secur Netw 2016;11(1/2):3–11.
[163] Suresh VM, Sidhu R, Karkare P, Patil A, Lei Z, Basu A. Powering the IoT through embedded machine learning and lora. In: 2018 IEEE 4th world forum on internet of things. IEEE; 2018, p. 349–54.
[164] Lei L, You L, Dai G, Vu TX, Yuan D, Chatzinotas S. A deep learning approach for optimizing content delivering in cache-enabled HetNet. In: 2017 International symposium on wireless communication systems. IEEE; 2017, p. 449–53.
[165] Jamal MS, Ochiai H, Esaki H, Kataoka K, et al. Instruct: A clustering based identification of valid communications in iot networks. In: 2018 Fifth international conference on internet of things: systems, management and security. IEEE; 2018, p. 228–33.
[166] Samarakoon S, Bennis M, Saad W, Latva-aho M. Dynamic clustering and on/off strategies for wireless small cell networks. IEEE Trans Wireless Commun 2015;15(3):2164–78.
[167] Tabrizi H, Farhadi G, Cioffi JM. Casra: An algorithm for cognitive tethering in dense wireless areas. In: 2013 IEEE global communications conference. IEEE; 2013, p. 3855–60.
[168] Duan S, Chen K, Yu X, Qian M. Automatic multicarrier waveform classification via PCA and convolutional neural networks. IEEE Access 2018;6:51365–73.
[169] Hussain F, Anpalagan A, Khwaja AS, Naeem M. Resource allocation and congestion control in clustered M2M communication using Q-learning. Trans Emerg Telecommun Technol 2017;28(4):e3039.
[170] Liu R, Lee M, Yu G, Li GY. User association for millimeter-wave networks: A machine learning approach. IEEE Trans Commun 2020.
[171] Galkin B, Amer R, Fonseca E, DaSilva LA. Intelligent base station association for UAV cellular users: A supervised learning approach. In: 2020 IEEE 3rd 5G world forum. IEEE; 2020, p. 383–8.

[172] Kudo T, Ohtsuki T. Cell range expansion using distributed Q-learning in heterogeneous networks. EURASIP J Wireless Commun Networking 2013;2013(1):61.
[173] Li Z, Wang C, Jiang C-J. User association for load balancing in vehicular networks: An online reinforcement learning approach. IEEE Trans Intell Transp Syst 2017;18(8):2217–28.
[174] Pervez F, Jaber M, Qadir J, Younis S, Imran MA. Fuzzy Q-learning-based user-centric backhaul-aware user cell association scheme. In: 2017 13th International wireless communications and mobile computing conference. IEEE; 2017, p. 1840–5.
[175] Ding H, Zhao F, Tian J, Li D, Zhang H. A deep reinforcement learning for user association and power control in heterogeneous networks. Ad Hoc Netw 2020;102:102069.
[176] Chou P-Y, Chen W-Y, Wang C-Y, Hwang R-H, Chen W-T. Deep reinforcement learning for MEC streaming with joint user association and resource management. In: ICC 2020-2020 IEEE international conference on communications. IEEE; 2020, p. 1–7.
[177] Zhang Q, Liang Y-C, Poor HV. Intelligent user association for symbiotic radio networks using deep reinforcement learning. IEEE Trans Wireless Commun 2020.
[178] Zhang Q, Zhang L, Liang Y-C, Kam P-Y. Backscatter-NOMA: A symbiotic system of cellular and internet-of-things networks. IEEE Access 2019;7:20000–13.
[179] Long R, Guo H, Zhang L, Liang Y-C. Full-duplex backscatter communications in symbiotic radio systems. IEEE Access 2019;7:21597–608.
[180] Guo H, Liang Y-C, Long R, Xiao S, Zhang Q. Resource allocation for symbiotic radio system with fading channels. IEEE Access 2019;7:34333–47.
[181] Raikar MM, Meena S, Mulla MM, Shetti NS, Karanandi M. Data traffic classification in software defined networks (SDN) using supervised-learning. Procedia Comput Sci 2020;171:2750–9.
[182] Sun G, Liang L, Chen T, Xiao F, Lang F. Network traffic classification based on transfer learning. Comput Electr Eng 2018;69:920–7.
[183] Yu C, Liu Y, Yao D, Yang LT, Jin H, Chen H, et al. Modeling user activity patterns for next-place prediction. IEEE Syst J 2015;11(2):1060–71.
[184] Sinclair N, Harle D, Glover IA, Irvine J, Atkinson RC. An advanced SOM algorithm applied to handover management within LTE. IEEE Trans Veh Technol 2013;62(5):1883–94.
[185] Castro-Hernandez D, Paranjape R. Classification of user trajectories in LTE HetNets using unsupervised shapelets and multiresolution wavelet decomposition. IEEE Trans Veh Technol 2017;66(9):7934–46.
[186] Wu C-L, Chiu T-C, Wang C-Y, Pang A-C. Mobility-aware deep reinforcement learning with glimpse mobility prediction in edge computing. In: ICC 2020-2020 IEEE international conference on communications. IEEE; 2020, p. 1–7.
[187] Zhang H, Dai L. Mobility prediction: A survey on state-of-the-art schemes and future applications. IEEE Access 2018;7:802–22.
[188] Dang F, Yang X, Long K. Spray and forward: Efficient routing based on the Markov location prediction model for DTNs. Sci China Inf Sci 2012;55(2):433–40.
[189] Song C, Qu Z, Blumm N, Barabási A-L. Limits of predictability in human mobility. Science 2010;327(5968):1018–21.
[190] Monreale A, Pinelli F, Trasarti R, Giannotti F. Wherenext: a location predictor on trajectory pattern mining. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining. 2009, p. 637–46.
[191] Cicioğlu M. Performance analysis of handover management in 5G small cells. Comput Stand Interfaces 2021;75:103502.
[192] Ali Z, Baldo N, Mangues-Bafalluy J, Giupponi L. Machine learning based handover management for improved QoE in LTE. In: NOMS 2016-2016 IEEE/IFIP network operations and management symposium. IEEE; 2016, p. 794–8.
[193] Lee C, Cho H, Song S, Chung J-M. Prediction-based conditional handover for 5G mm-wave networks: A deep-learning approach. IEEE Veh Technol Mag 2020;15(1):54–62.
[194] Shubyn B, Lutsiv N, Syrotynskyi O, Kolodii R. Deep learning based adaptive handover optimization for ultra-dense 5g mobile networks. In: 2020 IEEE 15th international conference on advanced trends in radioelectronics, telecommunications and computer engineering. IEEE; 2020, p. 869–72.
[195] Aljeri N, Boukerche A. A two-tier machine learning-based handover management scheme for intelligent vehicular networks. Ad Hoc Netw 2019;94:101930.
[196] Wu J, Liu J, Huang Z, Zheng S. Dynamic fuzzy Q-learning for handover parameters optimization in 5g multi-tier networks. In: 2015 International conference on wireless communications & signal processing. IEEE; 2015, p. 1–5.
[197] Dhahri C, Ohtsuki T. Q-learning cell selection for femtocell networks: Single-and multi-user case. In: 2012 IEEE global communications conference. IEEE; 2012, p. 4975–80.
[198] Dhahri C, Ohtsuki T. Adaptive Q-learning cell selection method for open-access femtocell networks: Multi-user case. IEICE Trans Commun 2014;97(8):1679–88.
[199] Wang Z, Li L, Xu Y, Tian H, Cui S. Handover control in wireless systems via asynchronous multiuser deep reinforcement learning. IEEE Internet Things J 2018;5(6):4296–307.
[200] Muñoz P, Barco R, de la Bandera I. Load balancing and handover joint optimization in LTE networks using fuzzy logic and reinforcement learning. Comput Netw 2015;76:112–25.
[201] Goyal T, Kaushal S. Handover optimization scheme for LTE-advance networks based on AHP-TOPSIS and Q-learning. Comput Commun 2019;133:67–76.
[202] Yajnanarayana V, Rydén H, Hévizi L. 5G handover using reinforcement learning. In: 2020 IEEE 3rd 5G world forum. IEEE; 2020, p. 349–54.
[203] Koda Y, Yamamoto K, Nishio T, Morikura M. Reinforcement learning based predictive handover for pedestrian-aware mmwave networks. In: IEEE INFOCOM 2018-IEEE conference on computer communications workshops. IEEE; 2018, p. 692–7.
[204] Mollel MS, Abubakar AI, Ozturk M, Kaijage S, Kisangiri M, Zoha A, et al. Intelligent handover decision scheme using double deep reinforcement learning. Phys Commun 2020;101133.
[205] Guo D, Tang L, Zhang X, Liang Y-C. Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning. IEEE Trans Veh Technol 2020;69(11):13124–38.
[206] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[207] 36.300 GT. Evolved universal terrestrial radio access (E-UTRA) and evolved universal terrestrial radio access network (E-UTRAN); Overall description; Stage 2. 2011.
[208] Antipolis S. 3Rd generation partnership project; technical specification group services and system aspects; telecommunication management; self-organizing networks (SON) policy network resource model (NRM) integration reference point (IRP); information service (IS)(release 11). 2012, p. 1–41, 3GPP Draft.
[209] Li D, Zhang B, Li C. A feature-scaling-based k-nearest neighbor algorithm for indoor positioning systems. IEEE Internet Things J 2015;3(4):590–7.
[210] Wu Z, Fu K, Jedari E, Shuvra SR, Rashidzadeh R, Saif M. A fast and resource efficient method for indoor positioning using received signal strength. IEEE Trans Veh Technol 2016;65(12):9747–58.
[211] Van Nguyen T, Jeong Y, Shin H, Win MZ. Machine learning for wideband localization. IEEE J Sel Areas Commun 2015;33(7):1357–80.
[212] Wymeersch H, Maranò S, Gifford WM, Win MZ. A machine learning approach to ranging error mitigation for UWB localization. IEEE Trans Commun 2012;60(6):1719–28.
[213] Xu Y, Zhou M, Meng W, Ma L. Optimal KNN positioning algorithm via theoretical accuracy criterion in WLAN indoor environment. In: 2010 IEEE global telecommunications conference GLOBECOM 2010. IEEE; 2010, p. 1–5.
[214] Ye X, Yin X, Cai X, Yuste AP, Xu H. Neural-network-assisted UE localization using radio-channel fingerprints in LTE networks. IEEE Access 2017;5:12071–87.
[215] Xiao C, Yang D, Chen Z, Tan G. 3-D BLE indoor localization based on denoising autoencoder. IEEE Access 2017;5:12751–60.
[216] Mo Y, Zhang Z, Meng W, Agha G. Space division and dimensional reduction methods for indoor positioning system. In: 2015 IEEE international conference on communications. IEEE; 2015, p. 3263–8.
[217] Yoo J, Johansson KH, Kim HJ. Indoor localization without a prior map by trajectory learning from crowdsourced measurements. IEEE Trans Instrum Meas 2017;66(11):2825–35.
[218] Dai H, Ying W-h, Xu J. Multi-layer neural network for received signal strength-based indoor localisation. IET Commun 2016;10(6):717–23.
[219] Wang X, Gao L, Mao S, Pandey S. CSI-based fingerprinting for indoor localization: A deep learning approach. IEEE Trans Veh Technol 2016;66(1):763–76.
[220] Wang X, Gao L, Mao S. BiLoc: Bi-modal deep learning for indoor localization with commodity 5ghz wifi. IEEE Access 2017;5:4209–20.
[221] Doctor F, Hagras H, Callaghan V. A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. IEEE Trans Syst, Man, Cybern A: Syst Hum 2004;35(1):55–65.
[222] Acampora G, Loia V, Vitiello A. Distributing emotional services in ambient intelligence through cognitive agents. Serv Orient Comput Appl 2011;5(1):17–35.
[223] Garcia-Valverde T, Garcia-Sola A, Hagras H, Dooley JA, Callaghan V, Botia JA. A fuzzy logic-based system for indoor localization using WiFi in ambient intelligent environments. IEEE Trans Fuzzy Syst 2012;21(4):702–18.
[224] Klaine PV, Imran MA, Onireti O, Souza RD. A survey of machine learning techniques applied to self-organizing cellular networks. IEEE Commun Surv Tutor 2017;19(4):2392–431.
[225] Arjoune Y, Faruque S. Artificial intelligence for 5g wireless systems: Opportunities, challenges, and future research direction. In: 2020 10th Annual computing and communication workshop and conference. IEEE; 2020, p. 1023–8.
[226] Zhu H, Akrout M, Zheng B, Pelegris A, Jayarajan A, Phanishayee A, et al. Benchmarking and analyzing deep neural network training. In: 2018 IEEE international symposium on workload characterization. IEEE; 2018, p. 88–100.
[227] Weiss K, Khoshgoftaar TM, Wang D. A survey of transfer learning. J Big Data 2016;3(1):1–40.
[228] Li R, Zhao Z, Chen X, Palicot J, Zhang H. TACT: A transfer actor-critic learning framework for energy saving in cellular radio access networks. IEEE Trans Wireless Commun 2014;13(4):2000–11.

[229] Ahmed KI, Tabassum H, Hossain E. Deep learning for radio resource allocation in multi-cell networks. IEEE Network 2019;33(6):188–95.
[230] Lee W, Kim M, Cho D-H. Deep power control: Transmit power control scheme based on convolutional neural network. IEEE Commun Lett 2018;22(6):1276–9.
[231] Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning, Vol. 1, no. 2. MIT press Cambridge; 2016.
[232] Victor K. Video hosting service based in Beijing, China. 2003, [Online]. Available: https://www.youku.com/.
[233] Derpanis KG, Lecce M, Daniilidis K, Wildes RP. Dynamic scene understanding: The role of orientation features in space and time in scene classification. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE; 2012, p. 1306–13.
[234] Sun Y, Peng M, Mao S. Deep reinforcement learning-based mode selection and resource management for green fog radio access networks. IEEE Internet Things J 2018;6(2):1960–71.
[235] Feng M, Mao S. Dealing with limited backhaul capacity in millimeter-wave systems: A deep reinforcement learning approach. IEEE Commun Mag 2019;57(3):50–5.
[236] Xiao K, Mao S, Tugnait JK. TCP-drinc: Smart congestion control based on deep reinforcement learning. IEEE Access 2019;7:11892–904.
[237] Shi D, Ding J, Errapotu SM, Yue H, Xu W, Zhou X, et al. Deep *Q*-network-based route scheduling for TNC vehicles with passengers' location differential privacy. IEEE Internet Things J 2019;6(5):7681–92.
[238] Yu Y, Wang T, Liew SC. Deep-reinforcement learning multiple access for heterogeneous wireless networks. IEEE J Sel Areas Commun 2019;37(6):1277–90.
[239] Huang C, Hu S, Alexandropoulos GC, Zappone A, Yuen C, Zhang R, et al. Holographic MIMO surfaces for 6G wireless networks: Opportunities, challenges, and trends. IEEE Wirel Commun 2020;27(5):118–25.
[240] Tan Y, Yang J, Gopalakrishnan N. Self-learning, adaptive approach for intelligent analytics-assisted self-organizing-networks (SONs). 2019, US Patent 10, 382, 979.
[241] Feng S, Seidel E. Self-organizing networks (SON) in 3GPP long term evolution. Nomor Res GmbH 2008;20.
[242] Blasco P, Bennis M, Dohler M. Backhaul-aware self-organizing operator-shared small cell networks. In: 2013 IEEE international conference on communications. IEEE; 2013, p. 2801–6.
[243] Jaber M, Imran MA, Tafazolli R, Tukmanov A. A multiple attribute user-centric backhaul provisioning scheme using distributed SON. In: 2016 IEEE global communications conference. IEEE; 2016, p. 1–6.
[244] Cao G, Lu Z, Wen X, Lei T, Hu Z. AIF: An artificial intelligence framework for smart wireless network management. IEEE Commun Lett 2017;22(2):400–3.
[245] Amiri R, Almasi MA, Andrews JG, Mehrpouyan H. Reinforcement learning for self organization and power control of two-tier heterogeneous networks. IEEE Trans Wireless Commun 2019;18(8):3933–47.
[246] Silva MAL, de Souza SR, Souza MJF, Bazzan ALC. A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems. Expert Syst Appl 2019;131:148–71.
[247] Galindo-Serrano A, Giupponi L, Auer G. Distributed learning in multiuser OFDMA femtocell networks. In: 2011 IEEE 73rd vehicular technology conference. IEEE; 2011, p. 1–6.
[248] Chen M, Saad W, Yin C. Virtual reality over wireless networks: Quality-of-service model and learning-based resource management. IEEE Trans Commun 2018;66(11):5621–35.
[249] Samarakoon S, Bennis M, Saad W, Latva-aho M. Backhaul-aware interference management in the uplink of wireless small cell networks. IEEE Trans Wireless Commun 2013;12(11):5813–25.
[250] Chen X, Zhang H, Wu C, Mao S, Ji Y, Bennis M. Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning. IEEE Internet Things J 2018;6(3):4005–18.
[251] Klein A, Kuruvatti NP, Schneider J, Schotten HD. Fuzzy Q-learning for mobility robustness optimization in wireless networks. In: 2013 IEEE globecom workshops. IEEE; 2013, p. 76–81.
[252] Ma B, Guo W, Zhang J. A survey of online data-driven proactive 5G network optimisation using machine learning. IEEE Access 2020;8:35606–37.
[253] Saruta K, Kato N, Abe M, Nemoto Y. High accuracy recognition of ETL9b using exclusive learning neural network-< cd02d36. gif>(ELNET-< cd02d36. gif>). IEICE Trans Inf Syst 1996;79(5):516–22.
[254] Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, et al. Mastering the game of go without human knowledge. Nature 2017;550(7676):354–9.
[255] Paullada A, Raji ID, Bender EM, Denton E, Hanna A. Data and its (dis) contents: A survey of dataset development and use in machine learning research. 2020, arXiv preprint arXiv:2012.05345.
[256] Kahn G, Abbeel P, Levine S. Badgr: An autonomous self-supervised learning-based navigation system. 2020, arXiv preprint arXiv:2002.05700.

Sepehr Ashtari (Graduate Student Member, IEEE) received the B.S. degree in electrical and electronic engineering from Eastern Mediterranean University (EMU), North Cyprus, Turkey, in 2016 and the M.S. degree in telecommunication engineering from the University of New South Wales (UNSW), Sydney, Australia, in 2019. He is currently pursuing the Ph.D. degree in telecommunication and information technology at the University of Technology (UTS), Sydney, Australia.

From 2015 to 2016, he was a Research Assistant with the Department of Electrical and Electronic Engineering, North Cyprus, Turkey. He has worked as a research fellow at the Commonwealth Scientific and Industrial Research Organisation (CSIRO) on several wireless communication projects. His research interest includes improvement of wireless cellular communication, routing protocols, software or knowledge-based networking, 5G resource allocation, and machine learning optimization in wireless networks.

Mr. Ashtari's awards and honors include the Fellowship of Excellence from Department of Electrical and Electronic Engineering, the High Honors degree in B.S. as well as M.S., and Fellowship award from UTS Faculty of Engineering and IT.

Ian Zhou (Graduate Student Member, IEEE) received the B.S. degree in computer science from the University of Sydney, Australia, in 2016. He also received the MBA degree from the University of Technology Sydney in 2019. His research interest includes AI, IoT, cyber–physical system, and blockchain. Currently, he is pursuing a Ph.D. degree at the University of Technology Sydney researching on machine learning-based frost monitoring systems.

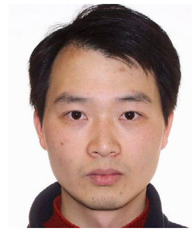
Mehran Abolhasan (Senior Member, IEEE) is currently an Associate Professor and the Deputy Head of the School of Electrical and Data Engineering, University of Technology Sydney. He has over 20 years of experience in research and development and serving in research leadership roles. Some of these previous roles include serving as the Director of research programs for the Faculty of Engineering and IT, and the Laboratory Director for the Telecommunication and IT Research Institute, University of Wollongong. He has authored over 160 international publications and has won over four million dollars in research funding. He won a number of major research project grants, including the ARC Discovery Project, ARC Linkage Project, and a number of CRC and other government and industry-based grants. He currently leads the Software-Defined Networks Lab at UTS and his current research interests include software defined networking, the IoT, wireless mesh, wireless body area networks, cooperative networks, 5G networks and beyond, and sensor networks.

Negin Shariati (Member, IEEE) is a Senior Lecturer in the School of Electrical and Data Engineering, Faculty of Engineering and IT, University of Technology Sydney (UTS), Australia. She established the state of the art RF and Communication Technologies (RFCT) research laboratory at UTS in 2018, where she is currently the Co-Director and leads research and development in RF-Electronics, Sustainable Sensing, Low-power Internet of Things, and Energy Harvesting. She leads the Sensing Innovations Constellation at Food Agility CRC (Corporative Research Centre), enabling new innovations in agriculture technologies by focusing on three key interrelated streams; Energy, Sensing and Connectivity.

Since 2018, she has held a joint appointment as a Senior Lecturer at Hokkaido University, externally engaging with research and teaching activities in Japan. She attracted over $850K worth of research funding across a number of CRC and industry projects, where she has taken the lead CI role and also contributed as a member of the CI team. Negin Shariati completed her Ph.D. in Electrical-Electronic and Communication Technologies at Royal Melbourne Institute of Technology (RMIT), Australia, in 2016. She worked in industry as an Electrical-Electronic Engineer from 2009–2012. Her research interests are in Microwave Circuits and

Systems, RF Energy Harvesting, low-power IoT, Simultaneous Wireless Information and Power Transfer, Wireless Networks, AgTech, and Renewable Energy Systems.

Justin Lipman (Senior Member, IEEE) received a Ph.D. in Telecommunications Engineering from University of Wollongong, Australia in 2004. He is an Industry Associate Professor at the University of Technology Sydney (UTS) and a visiting Associate Professor at Hokkaido University's Graduate School of Engineering. He is the Director of Research Translation in the Faculty of Engineering and IT and is Director of the RF Communications Technologies (RFCT) Lab — where he leads industry engagement in RF technologies, Internet of Things, Tactile Internet and Software Defined Communication. He serves as committee member in Standards Australia contributing to International IoT standards and Digital Twins. Prior to joining UTS, he was based in Shanghai, China and held a number of senior management and technical leadership roles at Intel and Alcatel driving research and innovation, product development, architecture and IP generation. He is an IEEE Senior Member. His research interests are in all "things" adaptive, connected, distributed and ubiquitous.



Wei Ni (Senior Member, IEEE) received the B.E. and Ph.D. degrees in Electronic Engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. Currently, he is a Group Leader and Principal Research Scientist at CSIRO, Sydney, Australia, and an Adjunct Professor at the University of Technology Sydney and Honorary Professor at Macquarie University. He was a Postdoctoral Research Fellow at Shanghai Jiaotong University from 2005–2008; Deputy Project Manager at the Bell Labs, Alcatel/Alcatel-Lucent from 2005 to 2008; and Senior Researcher at Devices R&D, Nokia from 2008 to 2009. His research interests include machine learning, stochastic optimization, online learning, as well as their applications to system efficiency and integrity.

Dr Ni is the Chair of IEEE Vehicular Technology Society (VTS) New South Wales (NSW) Chapter since 2020 and an Editor of IEEE Transactions on Wireless Communications since 2018. He served first the Secretary and then Vice-Chair of IEEE NSW VTS Chapter from 2015 to 2019, Track Chair for VTC-Spring 2017, Track Co-chair for IEEE VTC-Spring 2016, Publication Chair for BodyNet 2015, and Student Travel Grant Chair for WPMC 2014.