IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 13, 2021, accepted December 20, 2021, date of publication January 6, 2022, date of current version January 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140719

A Survey on Reinforcement Learning-Aided Caching in Heterogeneous Mobile Edge Networks

NIKOLAOS NOMIKOS[1], (Senior Member, IEEE), SPYROS ZOUPANOS[2], THEMISTOKLIS CHARALAMBOUS[3,4], (Senior Member, IEEE), AND IOANNIS KRIKIDIS[1], (Fellow, IEEE)

[1]IRIDA Research Centre for Communication Technologies, Department of Electrical and Computer Engineering, University of Cyprus, 1678 Nicosia, Cyprus
[2]Department of Informatics, Ionian University, 491 00 Corfu, Greece
[3]Department of Electrical and Computer Engineering, University of Cyprus, 1678 Nicosia, Cyprus
[4]Department of Electrical Engineering and Automation, School of Electrical Engineering, Aalto University, 02150 Espoo, Finland

Corresponding author: Themistoklis Charalambous (themistoklis.charalambous@aalto.fi)

This work was supported in part by the European Regional Development Fund, and in part by the Republic of Cyprus through the Research and Innovation Foundation under Project INFRASTRUCTURES/1216/0017 (IRIDA).

ABSTRACT Mobile networks experience a tremendous increase in data volume and user density due to the massive number of coexisting users and devices. An efficient technique to alleviate this issue is to bring the data closer to the users by exploiting cache-aided edge nodes, such as fixed and mobile access points, and even user devices. Meanwhile, the fusion of machine learning and wireless networks offers new opportunities for network optimization when traditional optimization approaches fail or incur high complexity. Among the various machine learning categories, reinforcement learning provides autonomous operation without relying on large sets of historical data for training. In this survey, reinforcement learning-aided mobile edge caching solutions are presented and classified, based on the networking architecture and optimization target. As sixth generation (6G) networks will be characterized by high heterogeneity, fixed cellular, fog, cooperative, vehicular, and aerial networks are studied. The discussion of these works reveals that there exist reinforcement learning-aided caching schemes with varying complexity that can surpass the performance of conventional policy-based approaches. Finally, several open issues are presented, stimulating further interest in this important research field.

INDEX TERMS 6G, edge caching, heterogeneous networks, machine learning, mobile edge networks, reinforcement learning.

I. INTRODUCTION

Today, the wide commercial roll-out of fifth generation (5G) networks has become a reality, better supporting enhanced mobile broadband (eMBB) services, ultra-reliable and ultra-low latency (URLLC) critical applications, and massive machine type communications (mMTC) in the context of the Internet-of-Things (IoT) [1], [2]. Moving forward, sixth generation (6G) networks are expected to materialize around 2030 and at that time, the International Telecommunication Union (ITU) predicts that the total mobile data traffic volume will exceed 5 ZB per month, a 670-fold increase from 2010 [3]. Meanwhile, mobile subscriptions will more than triple, reaching 17.1 billion, compared to 5.32 billion in 2010.

Such figures necessitate novel wireless network design approaches and the recent adoption of machine learning (ML) solutions, promises significant performance gains. ML-based techniques enable the communication networks to exploit the wealth of data in various mobile applications and interact with their environments in order to explore different actions and then, according to the observed reward, they adapt and exploit the actions yielding the highest reward for their next ventures. A technique facilitating the evolution to 6G communications is mobile edge computing (MEC) and caching, where computation-intensive tasks take place near data collection and popular contents are in close proximity

The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut.

to users [4]–[7]. In this way, centralized cloud-based computation is avoided, while the backhaul and fronthaul links are relieved from constant content fetching from remote web servers. Moreover, computational and communication delays are considerably reduced, facilitating the provision of low-latency applications.

At the same time, traditional non-learning-based techniques might fail, due to the dynamic nature of wireless environment involving a large number of parameters and constraints, exhibiting prohibitive complexity for online network optimization. In such cases, ML-aided MEC and caching can exploit the plethora of mobile data and answer the questions of where, when and what to cache, as well as which tasks should be computed at the edge [8]–[10]. As online ML-aided network optimization is of tremendous importance towards 6G evolution, this survey focuses on reinforcement learning (RL)-aided edge caching in heterogeneous networks (HetNets) where novel paradigms, such as fog-based radio access, cooperative communications and mobile ground and aerial access points (APs), have dramatically changed the networking landscape.

A. CONTRIBUTIONS

In recent years, integrating ML in wireless networks has increased their capability to achieve 6G targets. In addition, edge caching is an enabling technique to improve the overall wireless network performance, through offloading and reduced content fetching from remote locations. Considering the importance of edge caching in the context of 6G networks and the high interest in developing RL- and deep reinforcement learning (DRL)[1]-aided solutions, this survey provides an exhaustive list of RL-based edge caching solutions in heterogeneous mobile environments. Thus, it is the first survey, focusing on discussing the intricacies of the different networking architectures and targeted performance metrics on the operation of RL-aided edge caching. More specifically, our contributions include:

- The opportunities and challenges for edge caching, due to the introduction of network architectures with heterogeneous capabilities and requirements are presented, together with the main categories of policy-based caching strategies and optimization objectives.
- Different cache-aided HetNets are covered, including conventional fixed cellular topologies, as well as novel networking paradigms, relying on fog-based radio access, cooperative networks, and highly mobile vehicular and flying networks.
- The requirements of each category on the design of RL-aided caching solutions is thoroughly discussed, presenting various learning frameworks, such as distributed multi-agent learning and bandit-based approaches, including possible implementation caveats.
- In each networking architecture, RL solutions are classified, according to the performance metric that is targeted by their corresponding reward functions, i.e., energy, spectral and caching efficiency, delay and Quality-of-Experience (QoE). Furthermore, details on the performance evaluation and comparisons with other learning and non-learning approaches are given.
- Open issues are highlighted, stemming from the interplay of RL-aided caching with various wireless networking aspects, such as physical-layer and multiple access design, security and network volatility from ground and aerial vehicles.

Employing ML for edge caching purposes can provide near optimal performance at a low complexity, tackling high dimensionality problems, involving different mobile networking parameters. Contrary to learning categories, such as supervised learning exploiting training datasets or unsupervised learning relying on past experiences, in RL, appropriate objective functions are formulated, capturing the impact of rewards/penalties from choosing specific actions from the state space.

There have been several surveys focusing on either the synergy of ML and wireless networks [8], [11] or the gains of edge caching through traditional optimization approaches [12], [13]. Additionally, most works dedicate only a part on ML-aided edge caching and non-exhaustive lists of relevant works [14]–[16] with a few works focusing on ML-aided edge caching. The survey in [17] investigates ML-based proactive caching, highlighting the improvement in small cells and UAV-aided networks. However, the majority of the reviewed works are non-learning-based. Another survey studies DL for edge caching, presenting the major DL categories and caching principles [18]. Still, the survey mostly focuses on the DL operation in a more general manner while the discussion often lacks details on the networking environments and performance evaluation. The use of artificial neural networks (ANNs) for wireless network optimization is examined in [19], including the consideration of edge caching applications. However, the tutorial dedicates only one part for ANN-aided edge caching and includes a small number of relevant works. An overview of AI-based wireless edge caching, including supervised, unsupervised, reinforcement and transfer learning is provided in [20]. Various challenges are highlighted, such as the dynamic environment due to mobility and fading. Still, a broad view of RL-based solutions that can handle the volatility of HetNets is not provided. Finally, the survey in [21] presents ML-based edge caching, providing a comprehensive taxonomy, according to the adopted machine learning technique, caching strategy, i.e. policy, location, and cache replacement, and the type and content delivery strategy. Although the included taxonomy offers a spherical view on the role of ML in cache-aided networks and a thorough comparison of various ML solutions, only a small part of the survey is dedicated for discussing RL-based solutions.

[1] DRL combines RL and neural network based function approximators in order to tackle the curse of dimensionality. See also discussion in III-C.

Table 1 summarizes the contributions and the scope of surveys, regarding RL-aided edge caching.

B. ORGANIZATION

The structure of this survey is as follows. First, Section II presents an introduction to various ML categories, highlighting their advantages and the role of RL towards autonomous network operation without needing huge training datasets. Then, Section III includes a taxonomy of RL-aided edge caching elements, providing their definitions and impact. The next five sections focus on fixed and moving edge networks, and for each one, caching schemes are classified, according to their performance target. More specifically, in Section IV, we present RL-aided edge caching studies in single- and multi-cell networks, while Section V focuses on low-complexity for radio access network (F-RAN) topologies. Then, Section VI includes cooperative caching approaches and local caching at mobile devices, in the context of device-to-device (D2D) communications. Subsequently, highly mobile and flexible networks are discussed in Section VII and Section VIII, i.e., vehicular and UAV-aided networks, respectively. Open issues in the area of RL-based edge caching are presented in Section IX, comprising among others, physical-layer issues and security concerns, as well as volatile networking architectures. Finally, conclusions are given in Section X. Overall, the structure of this survey is depicted in Fig. 1 and a list of acronyms is given in Table 2.

II. MACHINE LEARNING

Research on employing machines to process large data volumes, stemming from previously allocated tasks or simulated scenarios, towards learning to handle future tasks, has led to the tremendous growth of machine learning (ML) [22]. In mobile communication networks, a massive number of users and devices enjoy a broad range of services with different service requirements from HetNet nodes, having varying hardware capabilities. This explosive increase of wireless traffic demands highly complex network optimization solutions, posing difficulties to resource allocation of bandwidth, power and storage. The adoption of online ML solutions in such challenging settings leads to self-adaptive networks and accurate prediction of communication parameters, abiding to dynamic wireless conditions [23]. In this way, network performance will be enhanced, offering improved Quality-of-Service (QoS) and resource efficiency.

ML is mainly classified into three different categories; namely, supervised learning, unsupervised learning, and reinforcement learning (RL) [24]; see, Fig. 2. In a finer categorization, one can find semi-supervised learning, deep learning and more recently, federated learning [25] and transfer learning [26]. In what follows, we provide details on each of these classes.

- **Supervised learning:** In supervised learning, the algorithms rely on datasets, providing both the input and the output. Even though supervised learning provides improved decision-making, the need for labeled data might be prohibitive in practice. Algorithms in this category include classification and regression analysis which can facilitate the characterization of data traffic and content popularity.
- **Unsupervised learning:** Unsupervised learning approaches rely on training data that do not include labeled output. Clustering is a popular method to develop unsupervised learning algorithms, enabling pattern identification in datasets. In edge caching, users can be clustered based on, for example, their desired contents, mobility and willingness to cooperate with each other.
- **Semi-supervised learning:** An intermediate approach regarding the nature of the available data has been followed with semi-supervised learning. In this type of learning, both labeled and unlabeled data are exploited for training.
- **Reinforcement learning:** In RL, an agent's strategy is determined in an autonomous manner by considering the cost and reward of each action. Therefore, the main idea of this type of learning is radically different, as compared to the previous mentioned ones, which exploit historical data. Instead, RL algorithms are trained by using feedback on previously taken actions, adapting their behavior to the environment. In the edge caching case, various algorithms are used, such as Q-learning for predicting content request probability or deriving the popularity distribution. While in supervised learning the model is trained with the correct answer, in RL there is no answer but the reinforcement agent makes the decision how to perform a given task. If there does not exist any training dataset, RL learns from its experience. Hence, unlike other approaches, RL is about taking suitable action to maximize a reward (e.g., best possible behavior or path) in a particular situation.
- **Deep learning:** Deep learning (DL) is closely related to the above classes of ML. It relies on multiple layers to form artificial neural network architectures for accurate decision-making. In this hierarchical architecture, lower-level features define higher-level ones, while feature extraction is autonomously performed. In edge caching cases, DL can provide near-optimal policies for content placement and pushing without excessive complexity, even though large volumes of training data should be available. Here, the observation of the mobile edge environment leads to the formation of specific states that act as input to the deep neural network (DNN) for deciding the action that should be selected by the agent. Each action results in specific rewards, that in the long-term determine the efficiency of the DL policy.
- **Federated learning:** This approach decouples model training from requiring direct access to raw training data. In federated learning (FL) users exploit shared models trained from excessive amounts of data, without the need to centrally store it [25]. Here, devices take part as clients in a federation aiming at solving the learning task while being coordinated by a central server. Each client

TABLE 1. List of surveys presenting reinforcement learning-aided edge caching.

Reference	Short description	Scope of RL-aided edge caching in mobile networks
Anokye et al. [17]	A short survey on edge caching in small cell and UAV-based networks	- Studying the improvement in small cells and UAV-based cache-aided networks - The majority of the discussed works are non-learning-based
Wang et al. [18]	Study of DL for edge caching, presentation of major DL categories and caching principles	- Focus is given to DL operation in a general manner - Details on heterogeneous networking environments and performance improvement are missing
Chen et al. [19]	Architecture and use of artificial neural networks (ANNs) for wireless network optimization	- Various wireless networking problems are studied, including edge caching - Only a part of the survey is dedicated to ANN-aided edge caching, discussing a small number of works related to the application of RL solutions
Sheraz et al. [20]	Overview of AI-based wireless caching, including supervised, unsupervised, reinforcement and transfer learning	- RL-based solutions represent a subset of the discussed works - Classification is based on the performance target and not on the type of mobile network
Shuja et al. [21]	Presentation of various ML-based techniques for edge caching and relevant taxonomy	- The taxonomy gives a wide perspective on the role of ML in cache-aided networks - Comparison of the impact of ML techniques for improving edge caching performance - Only a small part of this work discusses RL-based solutions
This survey	RL-aided edge caching solutions and their impact on the performance of heterogeneous mobile networking environments	- A thorough overview of the integration of RL-aided edge caching in different mobile networking architectures is provided - For each mobile architecture, RL solutions are categorized and discussed depending on their performance target - Open issues focusing on various aspects of RL-aided edge caching, starting from the physical-layer up to volatile networking architectures

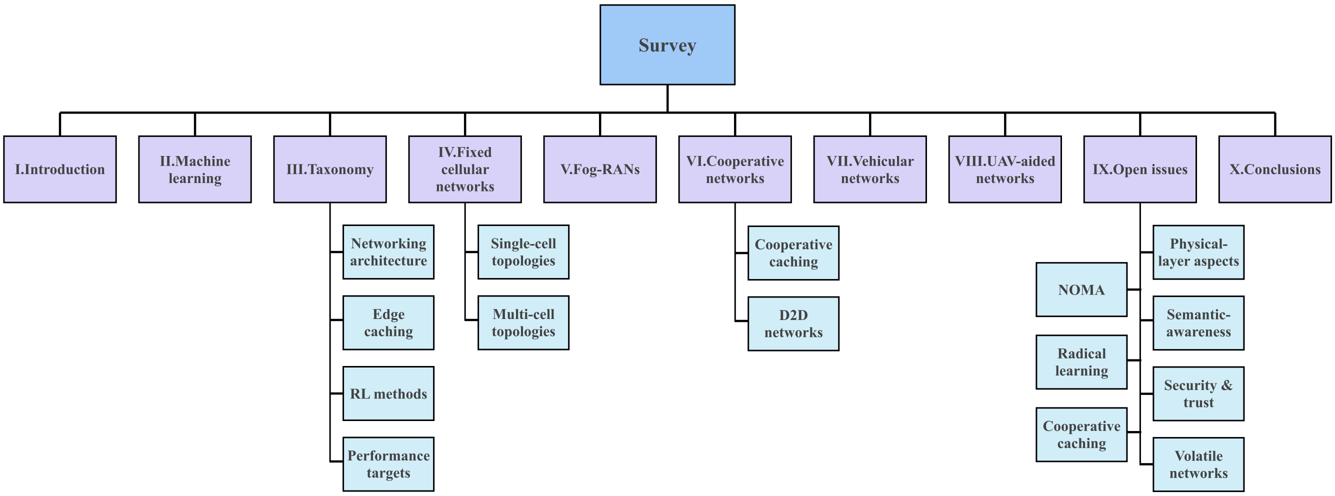

FIGURE 1. Survey structure.

maintains a local training dataset that is not uploaded to the server and only computes and communicates an update to the current global model of the server. FL benefits applications where training can be based on already available data at each client, and guarantees high privacy and security levels, since attacks affect only individual devices, and not the cloud. In mobile edge settings, FL-based model integration facilitates the construction of global popularity prediction models based on local models [27].

TABLE 2. List of acronyms.

Acronym	Definition	Acronym	Definition	Acronym	Definition
A3C	Asynchronous advantage actor-critic	AC	Actor-critic	AoI	Age of information
AP	Access point	BBU	Baseband unit	BLA	Bayesian learning automata
BS	Base station	C-RAN	Cloud radio access network	CDM	Content delivery market
CNN	Convolutional neural network	CPU	Central processing unit	CRNN	Convolutional recurrent neural network
D2D	Device-to-device	DCA	Deterministic caching algorithm	DDPG	Deep deterministic policy gradient
DDQN	Double deep Q-network	DL	Deep learning	DNN	Deep neural network
DQL	Deep Q-learning	DQN	Deep Q-network	DRL	Deep reinforcement learning
DTS	Double time-scale	ELSM	Echo liquid state machine	eMBB	Enhanced mobile broadband
EMQRN	External memory-based recurrent Q-network	F-RAN	Fog radio access network	FIFO	First-in first-out
FL	Federated learning	HetNet	Heterogeneous network	ICRP	Individual content request probability
IoT	Internet-of-Things	ITU	International telecommunication union	KNN	K-nearest neighbors
LFM	Least frequently caching and matching	LFU	Least frequently used	LMP	Local most popular
LRU	Least recently used	LSTM	Long-short-term memory	M2M	Machine-to-machine
MAB	Multi-armed bandit	MDP	Markov decision process	MEC	Mobile edge computing
MIMO	Multiple-input multiple-output	MINLP	Mixed integer non-linear program	ML	Machine learning
mMTC	Massive machine type communications	MNO	Moblie network operator	MVNO	Mobile virtual network operator
NOMA	Non-orthogonal multiple access	PLS	Physical-layer security	PPO	Proximal policy optimization
QLCCA	Q-learning collaborative cache algorithm	QoE	Quality-of-Experience	QoS	Quality-of-Service
RAN	Radio access network	RB	Radio bearer	RL	Reinforcement learning
RLNC	Random linear network coding	RRH	Remote radio head	RRM	Radio resource management
RSU	Road Side Unit	SAE	Stacked auto-encoder	SDDPG	Supervised deep deterministic policy gradient
SDN	Software-defined network	SNR	Signal-to-noise ratio	UAV	Unmanned aerial vehicle
UCB	Upper-confidence bound	URLLC	Ultra-reliable and ultra-low latency	V2I	Vehicle-to-infrastructure
V2V	Vehicle-to-vehicle	V2X	Vehicle-to-everything	VFA	Value function approximation
VR	Virtual reality				

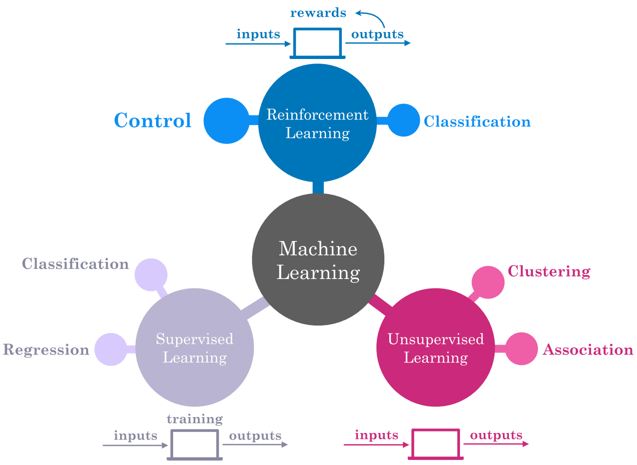

FIGURE 2. Different classifications of ML.

- **Transfer learning:** In edge environments, the energy and resource demands for model training might be prohibitive when constrained devices are involved. In such cases, knowledge transfer can enhance the learning performance without excessive data-labeling procedures. Thus, the transfer learning paradigm initially trains a base network, referred to as the "teacher" network and then, the learned features are transferred to a target "student" network [26]. In this way, the acquired knowledge from a general source problem is exploited to solve a related specific problem. Considering the edge devices as "students", transfer learning can provide significant resource savings, as long as the relation among the source and target problems is high.

III. TAXONOMY OF RL-AIDED CACHING IN HETNETS

Nowadays, conventional cellular architectures struggle to provide throughput and delay guarantees and a radical departure is currently taking place, exploiting cloud computing and the existence of edge nodes [28]. Cloud computing offers abundant processing power for tasks, such as baseband processing for cloud radio access networks (C-RANs), IoT applications with a massive number of sensors and mobile big data exploitation for network optimization [29], [30]. Unfortunately, centralized cloud architectures increase backhaul usage and end-to-end latency that might be intolerable for critical and real-time applications. So, bringing computation closer to the edge has been proposed [5], [31]. For this purpose, the caching capabilities of edge nodes allow the contents to be in proximity to the users, offering latency minimization, throughput maximization, backhaul offloading, reduced operational expenditure due to energy savings, and finally, extended lifetime to mobile terminals and IoT devices [6].

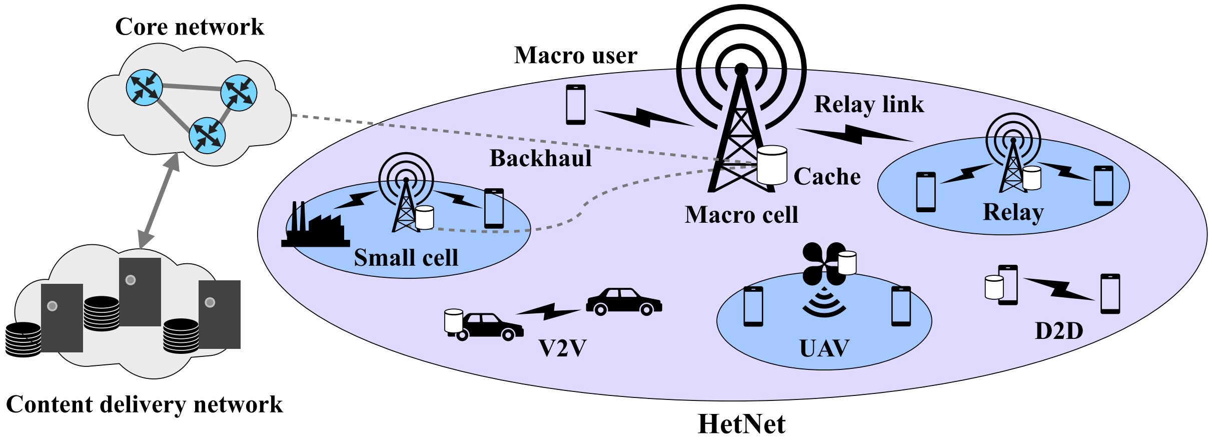

FIGURE 3. Different cases of edge caching in HetNets.

A. NETWORKING ARCHITECTURE

The MEC architecture requires from edge nodes to be equipped with storage capabilities for caching popular contents and avoiding constant fetching from remote web servers. Meanwhile, the wide range of different edge node types constitutes a challenging environment for optimizing the network performance [32].

An illustrative architecture is depicted in Fig. 3, showing different cases of edge caching in HetNets. More specifically, within the coverage area of a macro cell, a small cell caches content and serves users, requiring reliable and high throughput access, while a relay caches content that was transmitted from nearby users in the uplink and the macro BS in the downlink. Meanwhile, UAVs provide coverage to remote areas with limited coverage and store content that is scheduled for transmission towards the macro BS at a later moment, in order to reach the core network. Furthermore, various ad hoc communication paradigms exist, including cache-enabled devices communicating with each other, adopting D2D cooperation, as well as vehicle-to-vehicle (V2V) communication in highly mobile environments.

1) FIXED NETWORKS

The majority of networking architectures relies on fixed BSs, being designed towards increasing the wireless coverage, taking into account user density and mobility. Here, various categories exist.

a: CELLULAR NETWORKS

Multi-tier cellular topologies provide high frequency reuse, significantly increasing network capacity. Equipping BSs of different tiers with caches can further benefit network performance, as bottlenecks experienced in the backhaul are alleviated while lower end-to-end delay is guaranteed. Moreover, caching decisions can timely be exchanged among BSs through standardized X2/Xn interfaces.

b: F-RANs

Novel networking architectures, comprising F-RANs rely on low-complexity fog access points (F-APs), instead of conventional BSs, where only a part of baseband processing takes place at the F-APs [33]. F-RANs allow for flexible and low-cost network roll-out with improved coverage to edge nodes. Here, caching schemes should aim at alleviating fronthaul capacity constraints in order to avoid bottlenecks and guarantee high F-RAN performance.

c: COOPERATIVE NETWORKS

Further performance gains to cellular networks are offered through cooperative paradigms. More specifically, intelligent caching schemes can exploit the caching resources at different nodes, distributing the content for increased efficiency and robustness when network nodes experience outages. Also, cooperation between users through D2D communication can improve physical-layer aspects, such as coverage and transmit diversity while using storage at the devices for BS offloading [34].

2) MOVING NETWORKS

The introduction of highly mobile network nodes, providing wireless access and storage with increased flexibility represents another important field for edge caching.

a: VEHICULAR NETWORKS

The advent of autonomous driving and demand for improved road safety and in-car entertainment have led to the development of V2V and vehicle-to-everything (V2X) networking. In vehicular networks, communication takes place between vehicles, often in a multi-hop manner while topologies are highly dynamic [35]. In addition, the use of fixed road-side units (RSUs) facilitates connectivity and reduces instances of intermittent connectivity.

b: AERIAL NETWORKS

Another radical paradigm that has attracted significant attention is the use of flexible unmanned aerial vehicles (UAVs) to complement ground-based networks. UAV-aided networks offer fast recovery after disasters and emergency situations, on-demand capacity provisioning and coverage in remote and rural areas, enabling various IoT use cases, such as precision agriculture and fleet management [36].

B. EDGE CACHING

Data volume in mobile networks is exponentially increasing, as more users and IoT devices are connected and new services are developed. Characteristics, such as ultra-low latency and very high throughput cannot be guaranteed by traditional network architectures where remote servers or macro base stations (BSs) provide user content. Towards this end, edge caching represents an efficient approach to avoid overloading at specific network locations and constant data fetching which increases the load at the backhaul and fronthaul links. In this section, edge caching aspects are presented, starting from the heterogeneous characteristics of 6G networks. Then, different content update strategies are discussed and finally, details on a variety of performance targets are given.

1) CACHING LOCATION HETEROGENEITY

Modern wireless networks consist of network nodes with different capabilities, in terms of processing power, energy, storage, coverage and mobility. As a result, designing edge caching solutions should take into consideration these aspects when determining the caching location.

a: MACRO/MICRO/PICO/FEMTO BSs

In cellular networks, BSs at different tiers are exploited for bringing content closer to the users. Here, caching is performed not only at macro BSs but also at small BSs, such as micro, pico and femto BSs, providing opportunities for distributed caching and cooperation among BSs [12].

b: F-APs

As cloud and fog capabilities are being incorporated into the wireless architecture, low-complexity APs can be deployed in dense topologies. In such networks, simplified edge nodes in the form of F-APs provide storage and computing resources while being partly responsible for the baseband processing [33].

c: COOPERATIVE RELAYS

In cooperative networks, efficient caching can be performed where the cache status of different nodes is shared and data is cached across distributed buffers. In these networks, wireless relays enhance the flexibility in network deployment and improve the quality and reliability of wireless transmissions due to increased diversity and reduced path-loss [37]. At the same time, exploiting the relays' buffers improves the delay performance by avoiding data fetching and the resource scheduling efficiency when buffers are kept non-empty and non-full [38], [39].

d: USER DEVICES

Caching at the user devices exploits another dimension of cooperative networks, i.e., D2D communication paving the way for proximity services, performance gains to end users, high operation cost reduction to mobile network operators (MNOs).

e: GROUND VEHICLES/ROAD SIDE UNITS

In vehicular networks, mobility-aware caching can take place at the vehicles and at infrastructure-based RSUs [40]. Determining the caching location in these networks is highly challenging, as coverage and user association are subject to frequent changes.

f: UAVs

In aerial networks, the capabilities of the UAVs to optimize their trajectory or re-position to improve coverage increases the degrees of freedom for determining the caching policy. Cache-aided UAVs can be flexibly deployed in areas requiring wireless access, carrying with them user content and offering tremendous benefits to edge caching [36].

2) CACHE REPLACEMENT STRATEGIES

In mobile edge networks, there exist long-term characteristics that affect not only the caching location but also the caching strategy that should be employed. Such parameters include networking architecture, user and AP mobility, network traffic load, content popularity and QoS requirements. Still, the dynamic nature of mobile networks requires cache updates at shorter intervals using various policy-based strategies that have been proposed for Web content caching or novel paradigms integrating learning-based solutions.

a: REACTIVE CACHING

When a file is requested, reactive caching will decide whether or not it will be cached, according to criteria, such as file popularity, file size or remaining cache capacity. For example, a popular file will remain at the cache of a network node for a longer period of time.

b: PROACTIVE CACHING

In this approach, accurate content popularity prediction is vital in order to determine if a file will be popular in upcoming time periods. For this purpose, historical user preferences and content request data can be exploited. Proactive caching can mitigate the impact of network load variations by pre-fetching files and pushing them at cache-aided edge nodes.

c: POLICY-BASED

The main cache update parameters consist of recency (i.e., the time since the last object reference), frequency (i.e., the number of object requests over a specific time-period), object size, fetching cost and the time since the last modification [41].

1) LRU: The most popular recency strategy is the least recently used (LRU), relying on the temporal and spatial reference locality observed in content requests. LRU removes the least recently referenced object from the cache. LRU variations include EXP1, measuring object importance, as the elapsed time since the last object request [42] and LRU-Threshold, deciding not to cache an object if its size exceeds a pre-defined threshold [43].

2) LFU: Regarding frequency-based cache update, the least frequently used (LFU) and its extensions consider different object popularities. LFU removes the least frequently requested object from the cache. The LFU-Aging variation proposes an aging effect for once very popular objects that are not being requested for a long period of time [44].

A thorough discussion on each approach, was provided in [45]. Recency-based strategies adapt to new popular objects and are more simple to implement. Frequency-based strategies operate better when content popularity does not dramatically change over a specific time period, but they are more complex to implement.

d: LEARNING-BASED

As conventional optimization might not be able to provide optimal caching strategies, ML-based approaches can be employed to handle the large number of parameters of heterogeneous mobile edge networks. Learning-based caching solutions can rely on various ML categories or even use solutions from different categories to address specific caching problems.

The performance of proactive edge caching is based on the accurate prediction of different communication characteristics and the adaptation to the network dynamics. For example, supervised learning can leverage the wealth of mobile data in order to provide the learning-based caching strategy with accurate content popularity prediction or user profiling results. In the next phase, RL can be used for online optimization where one or more agents interact with the environment, considering the reward of different actions, in terms of a single or multiple performance metrics, such as cache hit rate or delay reduction.

C. REINFORCEMENT LEARNING METHODS

1) FRAMEWORK

Before describing the different Reinforcement learning (RL) methods, we discuss the main ideas of the mathematical framework of RL, which is, in general, employed to solve any problem that can be cast as a Markov decision problem (MDP) and its variants. It provides a formalization of intelligent decision making that is powerful and combines two principles - dynamic programming and supervised learning - and addresses problems that neither of the two principles can address individually.

Traditional dynamic programming suffers from the curse of modeling (for systems governed from a large number of random variables, it is often hard to derive the exact values of the associated transition probabilities) and curse of dimensionality (for large-scale systems with multiple states, it is impractical to store these values). However, RL can generate near-optimal solutions to large and complex MDPs avoiding these curses.

Additionally, supervised learning requires *a priori* a set of questions with the right answers for training the system, which is not feasible in several dynamical systems. Unlike supervised learning, RL systems do not require explicit input-output pairs for training. Rather, the system is simply given a goal to achieve and it then learns how to achieve that goal via trial-and-error interactions with a (possibly) dynamic environment.

2) ELEMENTS

RL problems consist of the following fundamental parts: a) (a model of) the environment, b) the reinforcement function (or reward signal), c) the policy, and d) the value function.

Model-free methods are explicitly trial-and-error learners. Here, the *environment* must be at least partially observable by the system. If the environment can be observed perfectly, then the system chooses actions based on true "states" of the environment. Model-based methods use models for learning and planning, as models allow inferences to be made about how the environment will behave.

After performing an action in a given state, the RL agent will receive some *reinforcement* (reward) in the form of a scalar value. The goal of the RL agent is to learn to perform actions that will maximize the sum of the reinforcements received over the long run. The reinforcement function therefore defines what the good and bad events for the agent are.

The *policy* determines which action should be performed in each perceived state of the environment. Therefore, the policy is a mapping from the perceived states to actions.

The *value function* is a mapping from states to state values and deals with how the RL agent learns to choose appropriate actions, or even how they might measure the utility of an action. It can be approximated using a function approximation. At the beginning, as it is expected, the function approximation of the optimal value function is not good. This means that the mapping from states to state values does not correspond to the actual one. Thus, the primary objective of learning is to find the correct mapping, from which the optimal policy can be extracted.

While rewards are provided directly by the environment, values must be estimated and re-estimated from the sequences of observations an agent makes over time. The different methods of RL are concerned with the choice of the function approximation of the value functions for efficiently estimating values. The reinforcement learning methods will be discussed next.

3) METHODS

We can divide the RL methods into two main categories:

- the *tabular solution methods*, in which the state and action spaces are small enough for the approximate value functions to be represented as arrays, or tables. In this case, the methods under this category can often find the optimal value function and the optimal policy.
- the *approximate solution methods*, which only find approximate solutions, but which in return can be applied effectively to much larger problems.

There are three main classes of tabular solution methods for solving Markov decision problems:

- dynamic programming (DP)
- Monte Carlo (MC)
- temporal-difference (TD) learning

These classes can be combined to extract the best features of each of them. Details can be found in [46].

When the number of states becomes enormous, the tabular solution methods are no longer suitable and the approach changes to finding a good approximate solution using limited computational resources. The main idea is to use a limited subset of the state space and generalize to produce a good approximation over a larger subset. Function approximation is an instance of supervised learning and can be used to approximate the optimal policy or to approximate value function (although they may be much more efficient if both the value function and the policy are approximated). RL with function approximation by deep ANNs is called "Deep RL" (DRL) and many impressive developments, as in mobile edge caching, have used DRL.

D. PERFORMANCE TARGETS

One important aspect of RL-aided edge caching is the performance metric that is targeted in the corresponding reward functions. As a result, in this survey, apart from categorizing each work according to the network type, we consider the main objective in their reward function, as a grouping criterion.

a: CACHE HIT RATIO

Various studies aim to improve the cache hit ratio which is defined as the ratio of cached files being requested by the end users over the total number of files that are stored in the cache. So, a high cache hit ratio corresponds to higher user satisfaction and backhaul/fronthaul offloading and thus, a more successful caching strategy.

b: SPECTRAL EFFICIENCY AND THROUGHPUT

Another important metric is related to the spectral efficiency, given in bps/Hz, corresponding to the achieved data rate over the available bandwidth. As more users and machines coexist in the network, spectral resources become more scarce and a higher frequency re-use is needed, e.g. by deploying small BSs, relays and mobile ground and aerial APs. Such heterogeneous topologies are complemented with caching at the edge nodes in order to reduce backhaul/fronthaul usage, bring data closer to the users and improve the throughput of the network and spectral efficiency.

c: END-TO-END DELAY

As ultra-low services are highly desirable in the context of the Tactile Internet and 6G networks, the end-to-end delay is a critical QoS parameter that determines the efficiency of an edge caching strategy. The reduction of end-to-end delay can be achieved on the one hand, by reducing or avoiding the number of hops that are needed to fetch the content from a remote server and on the other hand, by bring data closer to the end user and improve the quality of wireless transmission.

d: AGE OF INFORMATION

Recent studies, develop novel update scheduling strategies for minimizing the age of information (AoI) of the cached contents. In edge caching works, AoI quantifies how much time has passed from the moment that the current file version has been generated.

e: ENERGY EFFICIENCY

As the number of network nodes increases, achieving energy efficient network operation is important both for sustainable operation with reduced carbon footprint and reduced operational expenditure for the MNOs. The energy efficiency is usually measured in bits/Joule, i.e., the number of bits that are transmitted over the energy used for their transmission. Popular techniques to achieve high energy efficiency include power adaptation, switching off BSs, depending on the traffic pattern and offloading by bringing data in proximity to users through edge caching.

f: MOBILE VIRTUAL NETWORK OPERATOR REVENUE

When network slicing is adopted and the available spectrum is exploited by mobile virtual network operators (MVNOs), their revenue is determined by the amount of data that can be transmitted and the level of edge computing resources that are available. In such cases, the improvement of the MVNOs' revenue is targeted and the system reward is formulated, as a function of the access link signal-to-noise ratio (SNR), the MEC server computation capability, and the cache state.

g: QoE

In video streaming applications, a successful edge caching strategy is characterized by the user perceived QoE level. Various QoS parameters affect the QoE in video streaming services, such as jitter, packet delay and packet drop rate. Thus, the edge caching reward function for video application should consider as its objectives the accumulated reward by improving the performance of these metrics.

IV. FIXED CELLULAR NETWORKS

Cellular architectures comprising fixed BSs represent a major field where edge caching alleviates the burden of excessive data traffic from users within their coverage. Two networking architectures will be discussed, namely, single-cell and multi-cell multi-tier.

A. SINGLE-CELL TOPOLOGIES

Various works present RL-aided edge caching in single-cell networks that can be also integrated in more complicated environments, after necessary modifications.

1) ENERGY EFFICIENCY

Energy-awareness is an important issue in many edge caching use cases. In a cache-aided MEC networks, the paper in [47] studies task offloading, considering the joint optimization of cache, computation and power allocation for intensive computational tasks with stringent latency constraints. Initially, optimization is formulated as a mixed integer non-linear program (MINLP). Then, resource allocation is modeled as an MDP and a DRL framework is proposed, enabling the users and the AP to exploit historical data and increase the resource allocation efficiency. Furthermore, DRL provides a quasi-optimal solution with low-complexity, even under large MDP state space. From the simulations, it was shown that DRL reduces the energy consumption, as the AP caching capability increases, while for increasing computation capacity, the energy consumption performance is near-optimal. Also, comparisons with benchmarks without caching and different task computation strategies emphasize on the important energy gains of MDP-based DRL.

Another energy-aware solution has been investigated in [48], [49] where non-orthogonal multiple access (NOMA) has been employed. NOMA enables multiple users to simultaneously offload their tasks to APs, operating as edge computing servers and reducing the latency. Here, the caching of computational results reduces network traffic, as other users might request these results at a different time while enjoying the same application. In this context, the joint optimization of task offloading, computation resource allocation and caching decisions is solved through a long-short-term memory (LSTM) network. LSTM improves the exploration-exploitation trade-off when predicting task popularity. Resource allocation relies on a single-agent Q-learning algorithm while Bayesian learning automata (BLA) multi-agent Q-learning handles task offloading. Simulations depict the high prediction accuracy of LSTM. Comparisons of the single-agent Q-learning algorithm with three benchmarks, i.e., local computation at the mobile users, computation only at the AP and computation without caching highlight important energy savings for the RL algorithms for increasing caching and computation capacities. Finally, BLA multi-agent Q-learning reduces the energy consumption, compared to task offloading without BLA capability.

Focusing on energy-aware cache update strategy in small BSs with limited cache capacities, the works in [50], [51] consider a scenario with random resource availability and content requests. More specifically, time-varying and stochastic costs are assumed, being associated with file fetching from the cloud, incurring scheduling, routing and transmission costs. Also, cost includes memory and energy consumption due to caching at the small BS. Two cases are examined where in the first case, costs and content popularity follow known and stationary distributions, formulating a dynamic programming problem [52] that is solved through value-iteration-based RL. The second and more practical case considers limited cache capacity and unknown cost distributions and employs an online low-complexity Q-learning solver to determine the content update strategy. The caching versus fetching trade-off is evaluated for both cases with varying mean values for the caching and fetching costs. It is revealed that the online Q-learning without a priori knowledge of the statistical properties of the costs and content popularity offers almost the same average cost performance with value-iteration-based RL and improved cache-fetch decision-making in both stationary and non-stationary environments.

2) CACHING EFFICIENCY

Targeting to improve the data offloading and cache hit rate performance, the paper in [53] presented DRL with deep deterministic policy gradient (DDPG)-based training [54] and the Wolpertinger policy [55], relying on three entities. First, an actor function, receiving the cache state and the content requests, as inputs and providing a proto-actor from the set of valid actions. Then, K-nearest neighbors (KNN) mapping is employed, expanding the proto-actor to a set of valid actions from the action space. Finally, a critic function refines the actor for selecting the action with the highest Q-value from the expanded KNN set. Performance evaluation for centralized caching and comparisons with LRU, LFU, and first-in first-out (FIFO) policies reveal that actor-critic (AC) DRL can improve the cache hit rate in the short-term and avoid cache hit rate variations in the long-term.

Accurate cache update for a BS without content popularity knowledge is investigated in [56]. This problem is cast as an MDP where the BS cache status and the user requests represent the state space, while the decision of either keeping the current files or updating them define the action space. So, an LSTM and external memory-based recurrent Q-network (EMQRN)-based algorithm is developed to enhance the cache hit rate. Comparisons include LRU and FIFO without learning capabilities and the deep Q-network (DQN) algorithm of [57]. From the results, it can be seen that EMQRN leads to higher reward and faster convergence, compared to DQN. Still, the implementation of EMQRN in cooperative multi-cell topologies with cache status sharing remains an open problem.

3) QoE IMPROVEMENT

In a single caching server network transmitting short video content, a gradient-based DRL algorithm was developed in [58]. Two issues are jointly tackled, i.e., video quality selection and radio bearer (RB) control. The sequence of user content requests is modeled as an MDP for triggering

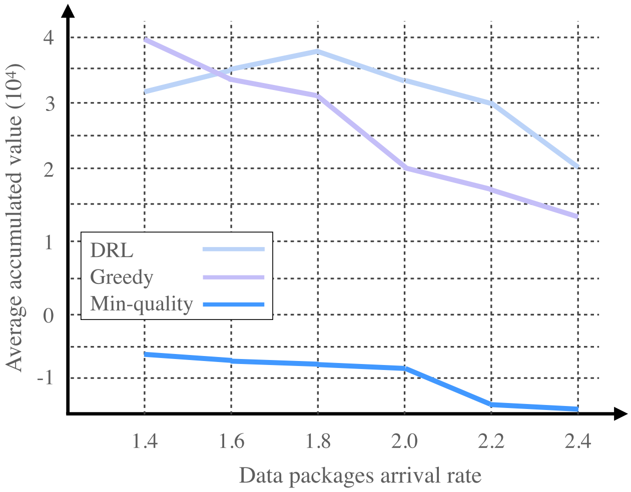

FIGURE 4. **Average accumulated value for varying packet arrival rate [58].**

RB control actions, such as setup, reconfiguration and release. For each action, the decision is made by considering the current state while training in settings with different parameters enables DRL to be employed in other settings, characterized by the same state space. Gradient method-based DRL is evaluated against a greedy policy, serving the request with the highest waiting time, and transmitting it at the highest video quality, and a minimum quality policy, serving the maximum number of requests at the lowest video quality. By considering the use of more RBs as cost and an increased video quality level as reward, different arrival rate cases are tested and, as it is observed in Fig. 4, DRL outperforms the greedy policy for rates between 1.8 to 2.4 while for rates below 1.6, the greedy method provides better performance, at a higher complexity. It is noted that generalizing the gradient-based DRL for more complex communication scenarios is an open issue.

Focusing on content-centric caching for improved QoE, the authors in [59] developed a DRL-based decision-making model, employing DNN for Q-value estimation. Optimization considered both latency and storage costs, outlining the negatively proportional relationship of the two metrics with QoE. As the network operates in a dynamic environment, DNN might not accurately estimate the Q-value and thus, fixed target network (F), experience replay buffer (E), and adaptive learning rate (L)-based DRL is proposed, leading to FEL-DRL. Fixed target network leads to stable convergence by using another neural network with fixed parameters which are periodically updated, according to the estimated network values. Meanwhile, experience replay avoids the temporal correlation of different training episodes, creating a dataset from the agent's experience and randomly using data batches for network training. Comparisons using Matlab˙ and TensorFlow showed that FEL-DRL achieves an average QoE score of 64, while DRL provided a score of 62 while other benchamrks, i.e. AC-DRL, FE-DRL and RL provided QoE values below 60.

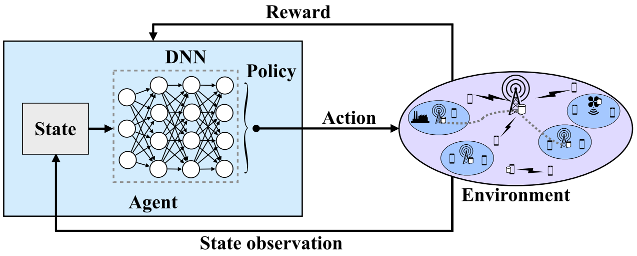

FIGURE 5. **Centralized deep learning architecture in the context of a two-tier HetNet environment.**

B. MULTI-CELL TOPOLOGIES

Rl-aided edge caching in multi-cell multi-tier networks has been investigated investigated in several works.

1) AGE OF INFORMATION REDUCTION

The authors in [60], [61] focus on cache update scheduling for age of information (AoI) minimization, in a two-tier HetNet where small cells act as content servers for dynamic content delivery, as shown in Fig. 5. AoI measures the elapsed time since the generation of the current file version [62], [63]. Content caching is formulated as a constrained MDP and enforced decomposition is employed for dynamic cache update. AoI is minimized by using multiple queues to monitor user requests. As the state space of the MDP subproblems can be large, DRL agents are trained for optimal cache update. Performance evaluation results, using PyTorch suggest that DDPG-based DRL offers improved convergence and reduced AoI. In the single dynamic content scenario, DDPG provides improved convergence, compared to DQN [57], while for multiple dynamic contents, the average AoI is reduced by 30% versus periodic update without considering the user request queues [64].

2) DELAY REDUCTION

The reduction of transmission delay and cache replacement cost in the long-term, in a two-tier small cell network is studied in [65]. Wireless coded caching is employed to distribute coded file fractions at different network nodes, thus avoiding the need for large caches. First, historical user file requests are exploited for predicting the future ones. In the next phase, a supervised deep deterministic policy gradient (SDDPG) approach based on both supervised learning and DRL solves the wireless coded caching problem. Aiming at accelerating the learning process, supervised learning is invoked to pre-train the neural network by solving an approximate cost minimization problem at each slot. Performance evaluation highlights that SDDPG reduces the total network cost, compared to short-term cost optimization. In addition, SDDPG exhibits a small performance gap, compared to the case of knowing the actual number of requests, serving as a performance upper-bound.

The MAB framework is adopted in [66], considering the transmission delay reduction over the case without caching, as the reward. Under unknown user preferences, the proposed collaborative caching schemes minimize the accumulated transmission delay over a finite time horizon. This work extends [67] which presented distributed and collaborative multi-agent MAB algorithms in stationary environments. Here, two stationarity cases are investigated for the file library and user preferences. For the stationary case, a fixed file set and time-invariant user preferences are considered and two high-complexity MAB algorithms are presented. Their regret performance is bounded by $\mathcal{O}(\log T_{\text{total}})$, where T_{total} denotes the total number of time-slots. Meanwhile, a lower complexity and distributed MAB solution is developed, considering that each small cell acts independently. In the stationary setting, an edge-based collaborative multi-agent MAB algorithm is proposed, relying on coordination graph edge-based reward assignment. Then, in the non-stationary case, the file set and user preferences dynamically vary and modified multi-agent MAB algorithms are given. More specifically, the exploration duration is reduced by assigning larger initial values to the actions of adding new content to the varying file set in each time-slot. Also, the upper confidence bound (UCB) terms are modified, as the small cells are unaware of the reward upper bound. Simulations show that the MAB algorithms reduce the delay, compared to LRU and LFU, and achieve a narrow performance gap, compared to a greedy algorithm for varying communication distance, cache size and mobility.

In [68] AC-based DRL joint user scheduling and content caching is proposed for delay minimization. In greater detail, the actor adopts stochastic caching, abiding to the Gibbs distribution and parameters are updated through gradient ascent by observing the environment states. The critic evaluates the actor policy and its rewards, in terms of delay, using DNN for value function approximation (VFA) and gradient estimation. The convergence of the AC-based DRL scheme is evaluated in a two-tier network for different actor and critic learning rates, highlighting that a low actor learning rate improves the convergence. Also, comparisons with AC-based DRL without caching and AC-based DRL without scheduling indicate 40% and 56% higher total rewards for the proposed caching scheme, respectively.

Joint delay and blockchain-based security optimization has been presented in [69], focusing on (machine-to-machine) M2M communication. As blockchain systems require increased computation time to complete the smart contracts, delay requirements might not be met [70]. So, system performance is enhanced through dueling DQN-based decision-making, regarding caching, computing and security. The dueling architecture allows DQN to efficiently learn the action value through the separate estimation of the state value and the reward of each action, leading to higher caching reward, reduced data computation overheads, and efficient blockchain processing. Performance comparisons against conventional DQN, a greedy-based strategy and random resource allocation for varying cache and block sizes, delay constraints and number of machine-type devices, showed reduced latency, and higher rewards for dueling DQN-based caching.

3) CACHING EFFICIENCY

AC-based DRL is employed in [53] for decentralized edge caching operation and improved cache hit rate performance. Comparisons with LRU, LFU and FIFO caching policies shows that AC-based DRL offers higher cache hit rate and reduced transmission delay by considering user location and enabling inter-cell communication in order to avoid caching the same content when coverage areas overlap.

The optimization of content caching and delivery policy under non-stationary content libraries through a user-assisted RL algorithm is the subject of [71]. The network utility consists of backhaul traffic offloading, cache hit rate, content retrieval and delivery. The learning-based algorithm exploits users' caches for offload the small cells during peak hours. The BSs' caches are divided in two parts where the first part stores new content from the users' caches, while the second part is used for content server updates. Content caching and delivery is formulated as a MAB problem, considering the spatio-temporal request dynamics. Also, the content library is modeled as a multi-arm system with unknown and stationary rewards. A central unit sequentially determines content caching, exploring possibly popular and rarely cached files and exploiting the empirical knowledge of caching content, yielding the highest rewards up to this point. MAB-based content caching and delivery operates in three phases, where in the first phase, content delivery takes place, then, in the second phase, one part of the small cell caches is updated with content from the users and finally, in the third phase, the other part of the caches is updated from the content server. Comparisons with benchmarks without a user-assisted phase highlights that the MAB-based algorithm is more robust against the spatio-temporal request variations and benefits from the use of the users' caches.

Distributed content placement for alleviating the traffic load from the backhaul infrastructure in a dense small cell network was studied in [72]. As the problem of optimal content placement is shown to be NP-hard, independently of knowing the file popularity. Thus, a learning-based coded caching solution is proposed where the small BSs learn the file popularity profiles by using the historical content request data. The learning framework considers the connectivity of users with the small BSs and relies on combinatorial MAB. The MAB-based learning framework can adapt to temporal content popularity variations, exploring the caching of new files and discovering their popularity versus exploring the caching of already-known high popularity files. For the reception of distributed cached contents, rateless coding is adopted, guaranteeing high decoding performance, as long as a specific fraction of the coded symbols has been received. Performance evaluation depicts that MAB-based distributed caching obtains higher rewards, compared to a local caching scheme neglecting the network connectivity status and uncoded caching.

The goal of [73] is to improve edge caching performance in networks where infrastructure providers lease their physical resources, in the form of BS storage and backhaul capacity to MVNOs. By investigating the joint optimization of cache leasing and content popularity prediction from the MVNOs' perspective for profit maximization, a Q-learning algorithm provides DL models with optimized hyper-parameters. The generated DL models are employed to predict the parameters of content popularity, i.e. future cache demand and request count. Using this information, the DL models compile lists with contents that should be cached at the BS. Performance evaluation focuses on the cache hit probability and backhaul usage and three different configurations for the unknown layer of the DL model, i.e., convolutional neural network (CNN), LSTM and convolutional recurrent neural network (CRNN). Feature selection results suggest that LSTM provides superior training and validation accuracy with reduced training time. Moreover, the best LSTM configurations and random caching are compared against the case without caching, showing a 16% cache hit probability improvement, compared to 12% by the random scheme, and a 17% backhaul usage reduction, compared to 12% by random scheme.

An extension to the RL-based meta-learning with enhanced searching space design and autonomous DL model generation of [73] with optimized hyper-parameters is given in [74], comprising two parts. In the first part, a cloud-based master meta-learner provides the DL models and decides which one to deploy. The second part involves a slave meta-learner located at each small BS, using the best DL model for popularity prediction after tuning its parameter through localized information. Simultaneously, the slave meta-learner provides prediction accuracy feedback to the master meta-learner, triggering the latter to explore a different model, in case of suboptimal performance. The RL-based meta-learning scheme is implemented, using Tensorflow [75] and Keras [76] and achieves a 10% and 30% cache hit rate improvement over the scheme in [73] and random caching, respectively.

MNO net profit maximization through DRL-based caching is presented in [77]. Here, contents are proactively pushed and cached at the users' devices, relying on RL for predicting the individual user behavior. Since the joint problem of proactive pushing and recommendation is characterized by large action and state spaces, a decomposition approach is followed. First, the recommendation subproblem focuses on increasing requests and providing revenue opportunities while the pushing subproblem targets transmission delay minimization. Considering the inter-dependency among the two subproblems, a double deep Q-network (DDQN), based on [78] is employed. Simulations are conducted to assess the performance of the dueling DDQN versus DDPG [54], advantage AC [79] and proximal policy optimization (PPO) [80]. Results highlight that dueling DDQN converges much faster while solving the recommendation sub-problem and provides the highest rewards. Even though, PPO provides almost the same reward as dueling DDQN, it requires around 43% additional training sessions. Meanwhile, the pushing policy exploits the user mobility pattern and the propagation characteristics, proactively pushing content under favorable channel conditions.

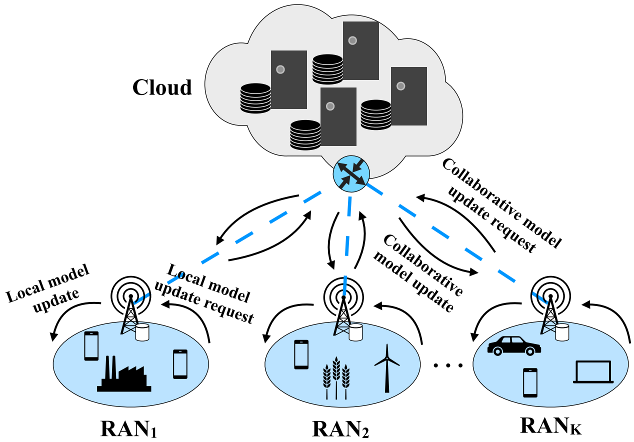
Cloud
Collaborative model update request
Collaborative model update
Local model update request
Local model update
RAN$_1$
RAN$_2$
RAN$_K$

FIGURE 6. Federated learning architecture for a multi-cell network, comprising *K* mobile edge RAN environments.

In [81], DRL-based network resource management for improved cache hit rate and computation offloading is presented. In mobile edge networks, the high amount of data, parameters and performance targets demands distributed DRL agent training. Here, two important aspects determine the appropriate distributed DRL architecture. First, although maintaining a DRL agent in every network node can provide improved performance, in practice, training will struggle due to differences in task load and network states, time constraints and data unavailability. Second, the distributed DRL architecture should overcome data imbalance and alleviate privacy concerns. So, FL is employed for distributed DRL agent training, reducing communication costs and offering improved privacy and security [25]. Fig. 6 shows a multi-cell network with K RANs, adopting FL for exploiting local model updates to improve the efficiency of the global collaborative model. The proposed In-Edge AI, avoids constant data uploading in the uplink, as FL relies on locally stored data and only calculates updates to the global model of the coordinating central node. Simulations compare DDQN with FL and centralized DDQN without FL, as well as LRU, LFU and FIFO. It is observed that DDQN with FL provides almost the same hit rate performance, as centralized DDQN and outperforms LRU, LFU, and FIFO. Moreover, since the simulated wireless topology is assumed to support the upload of the large amount of training data in the centralized DDQN approach, in practice, delay performance will be degraded. On the contrary, the small volume of data for FL-based training through the global model updates will slightly affect the delay.

In cases where user preferences and mobility patterns are unknown, the authors in [82] proposed a temporal-spatial recommendation policy addressing non-peaky local content popularity. This policy leads users to request their desired files in an efficient way, i.e., during specific time-slots and through appropriate BSs. Since a limited number of requests might hinder local popularity prediction, a Bernoulli mixture model is adopted to learn user preference and request probability. Then, the recommendation and caching policies are jointly optimized through RL. Nonetheless, this joint problem is characterized by large state and action spaces. Thus, it is decomposed into three subproblems, tackling user preference and file request probability estimation with or without recommendation, caching policy optimization, independently of recommendation, and finally, DRL-based recommendation policy optimization. Performance evaluation for a network with three BSs and a library of 200 files is presented. Benchmarks included random recommendation, no recommendation, global recommendation, based on the estimated global file popularity and local recommendation, based on aggregating the estimated local file popularity. It is shown that DRL overcomes individual user and aggregated preference estimation errors, better adapting the caching policy to user mobility through improved recommendation.

In [83], a multi-cell multiple-input multiple-output (MIMO) system is studied where the locations of the BSs are modeled through a Poisson point process (PPP). Given a content popularity profile, the average success probability of the system can be derived under the probabilistic caching assumption. The analysis of the average success probability showed that for inference limited systems, it is not affected by BS density. Furthermore, a Q-learning framework is developed to formulate the problem of dynamically learning the content placement strategies, targeting to maximize the average success probability and minimize the cache refresh rate. Through Q-function approximation the number of variables needed was reduced and simulations on a synthetic dataset showed that computation time can be reduced without affecting the performance.

4) SPECTRAL EFFICIENCY

Pushing and caching popular services in a three-tier network, consisting of broadcast BSs, cellular BSs and routers is studied in [84]. The broadcast BS is responsible for delivering the services and the caches of the routers are used to bring the content closer to the users. In case, a user does not receive its desired content, the router might act as a relay to establish connectivity with another router or a handover to a cellular BS takes place. Targeting the maximization of the equivalent network throughput, service scheduling is modeled as an MDP. Since the large state space entails excessive complexity, deep Q-learning (DQL) is used to derive the optimal policy. Towards addressing the large state space issue, the Q-function is approximated and modified using experience replay and target value update over several time steps. Performance comparisons with the algorithm of [85] and centralized caching with dynamic programming shows that for different popularity Zipf factors, DQL provides significantly higher equivalent throughput, especially for Zipf values below 1.5.

Edge caching through accurate content recommendation, based on personalized preferences is examined in [86]. Localized caching is presented, relying on the individual content request probability (ICRP) for content placement optimization and throughput maximization. This scheme is based on Bayesian learning, namely, constrained Bayesian probabilistic matrix factorization, considering the rating matrix imbalances, towards improving the prediction accuracy of unknown content ratings. This process facilitates the evaluation of personal preferences to obtain the ICRP. In the next step, RL exploits the ICRP and physical distance among caches and users for content placement, resulting in a deterministic caching algorithm (DCA). Further gains are provided through D2D cooperation for reduced delay and improved ICRP estimation. DCA is compared against random caching and probabilistic caching, in terms of root mean square error prediction, hit rate and system throughput. Comparisons reveal that DCA offers 90% increased throughput versus random caching, while D2D cooperation reduces the delay by 15.5% over the non-D2D case.

The work in [87] studies content placement at BSs for maximizing the average success transmission probability. The authors consider a network with BSs located, as a two-dimensional homogeneous PPP, while content popularity is assumed to be known and Zipf-distributed. The cost function is formulated using the average success probability and an online Q-learning approach is presented and evaluated for both small and large action spaces, depending on the number of cache size, content size and popularity profile set cardinality. Results indicate that for small action spaces, Q-learning converges to the optimal policy after 13 iterations while significantly more iterations are needed for convergence for large action spaces, i.e., around 2×10^3 iterations.

Aiming to mitigate the impact of tidal effects in mobile networks, i.e., increased network load in peak hours and low bandwidth utilization in idle periods, the authors in [88] propose proactive pushing and caching when the network is underutilized. This joint problem involves a transmission cost function, representing bandwidth fluctuation. Minimizing bandwidth fluctuation improves bandwidth utilization and energy efficiency and avoids duplicate data transmissions. So, hierarchical RL tackles the decomposed pushing and caching subproblems. The first subproblem is related to user cache optimization, employing Q-learning VFA. For the second subproblem, DRL is used to improve BS caching and tackle dimensionality issues. The performance of hierarchical RL is compared to policy-based schemes, such as LRU, LFU and local most popular (LMP), in three scenarios, i.e., caching at the BS, caching at the users, joint BS and user caching. It is observed that hierarchical RL outperforms the other policies in all three cases, while its advantage is significantly increased in the joint BS and user caching, efficiently utilizing the wired and wireless network.

The joint allocation of networking, caching, and computing resources in smart cities applications is examined in [89]. Assuming a dynamic virtualized networking environment where MVNOs manage multiple BSs, MEC servers and content caches, an excessive number of system states exists and traditional optimization faces difficulties in deriving the optimal policy. As a result, DRL is invoked, using DQN for Q-VFA, determining the resource allocation of networking, caching and computing resources. The reward function objective, corresponding to the MVNO's revenue consists of the access link SNR, the MEC server computation capability, and the cache state. Simulations, using TensorFlow evaluate DRL and alternative versions without caching, MEC offloading or virtualization. Results show that DRL offers a significantly higher total utility, independently of learning rate and the number of required central processing unit (CPU) cycles per task.

5) QoE IMPROVEMENT

Resource allocation and user association in a network providing live video streaming service is the subject of [90]. As the maximization of the QoE is prioritized, DDPG-based caching is presented, over traditional Lagrangian-based optimization. Initially, an optimization problem is formulated and shown to be non-linear and NP-hard. In order to convert it to a linear problem, binary decision variables are used, corresponding to BS caching content and user request of a specific video quality. In order to find a near-optimal solution, DDPG alternatively keeps one variable fixed and then, it relaxes both binary variables to be continuous. However, it is observed that in the user association/video quality subproblem, the sub-gradient method is inefficient and in some cases, only a locally optimal value might be obtained. So, DDPG is employed, first observing the state of resource utilization and determining the prices for each possible action. In the next step, the users are prompted to associate with the BSs, and request a specific video quality. In this way, the resource utilization is re-calculated based on the users' decisions and the QoE level is acquired as reward, facilitating the DDPG agent to evaluate the action and set appropriate NN weights. DDPG-based learning is evaluated against sub-gradient-based pricing [91] and the solution in [92]. It is concluded that DDPG offers higher QoE as the number of users increases, independently of resource availability.

The provision of enhanced QoE, while avoiding excessive energy consumption is studied in [93]. A software-defined network (SDN) is investigated, using a monitoring mechanism, processing several parameters related to BS cache, as well as user buffer status and video transmissions parameters. QoE and energy performance optimization is modeled, as a constrained MDP that is transformed into an unconstrained MDP by adopting the T-period drift-plus-penalty concept. The unconstrained MDP problem is tackled through asynchronous advantage actor-critic (A3C), employing its agents to run on a multi-core CPU with each thread processing one agent and providing a replica of the environment. Then, the globally shared parameter vector is asynchronously updated by using the cumulative gradients of multiple agents after a specific time period. Performance evaluation, using PyTorch and comparisons with DQN and traditional convex optimization are presented. A3C exhibits faster convergence than DQN and requires half the energy to provide the desired QoE. Moreover, for varying BSs and users numbers, A3C always outperforms DQN while convex optimization fails to follow network dynamics and falls behind both learning algorithms.

An additional work, focusing on QoE improvement and the reduction of latency for users requesting video content and backhaul usage is presented in [94]. Here, multi-agent AC DRL-based caching is developed, treating each network edge, as a cooperative learning agent and avoiding the large action spaces of centralized single-agent approaches. The proposed multi-agent collaborative caching (MaCoCache) enables each agent to consider not only its caching strategy but also, that of its neighbors, while relying on the AC algorithm. Also, to better adapt to network dynamics and exploit historical data, LSTM is integrated with MaCoCache. The proposed caching framework is compared against LRU, LFU and other learning-based alternatives, such as DRL without cooperation between agents and joint-action learners (JAL), utilizing stateless Q-learning-based caching [95]. It is revealed that MaCoCache offers 73%, 50%, 21% and 14% latency and 103%, 98%, 59%, 26% backhaul cost reduction versus LFU, LRU, DRL and JAL, respectively, as well as 13% and 7% improved edge hit ratio, compared to DRL and JAL, respectively.

Table 3 includes the studies on RL-aided caching in fixed cellular topologies, highlighting their main performance targets and the adopted RL approach.

Lessons learned: RL-aided edge caching in fixed cellular networks can provide improved performance over policy-based approaches, such as LRU, LFU and FIFO, independently of the considered reward function objective. Moreover, MAB-based RL can support practical scenarios without a priori knowledge of user preferences and mobility patterns. Also, there exist some works that developed RL-aided proactive content pushing and recommendation solutions [77], [88] but additional research is needed in this field, due to the significant offloading potential, alleviating the impact of network load variations. Another area that needs further efforts is related to SDN-based architectures, as only the work in [89] has proposed a relevant RL-aided edge caching solution. Overall, in multi-cell networks, distributed content caching operation must be better supported, as inter-cell coordination entails increased complexity. Moreover, FL approaches should be further studied, since maintaining DRL agents at each BS might be infeasible due to task load variations, time constraints, privacy issues and data imbalance. Finally, highly spectral efficient access schemes, such as user grouping for NOMA have not been extensively studied in conjunction with RL-aided edge caching.

V. FOG RADIO ACCESS NETWORKS

C-RANs provide flexible network deployment by relying on cloud-based centralized baseband units (BBUs) for signal

TABLE 3. List of works focusing on reinforcement learning-aided caching for fixed cellular networks.

Reference	Networking architecture	Performance target	RL solution
Yang et al. [47]	Single-cell	Energy consumption	DRL
Yang et al. [48], [49]	Single-cell NOMA	Energy consumption	BLA Q-learning
Sadeghi et al. [50], [51]	Single-cell	Energy, backhaul and storage costs	Value-iteration-based and Q-learning
Zhong et al. [53]	Single- and multi-cell	Cache hit rate	AC and KNN-based DRL
Wu et al. [56]	Single-cell	Cache hit rate	LSTM-based DRL
Wu et al. [58]	Single-cell	QoE, spectral efficiency	Gradient-based DRL
He et al. [59]	Single-cell	QoE	DRL
Ma et al. [60], [61]	Multi-cell	AoI	DDPG-based DRL
Zhang et al. [65]	Multi-cell	Delay, cache replacement cost	SDDPG RL
Xu et al. [66]	Multi-cell	Delay	Multi-agent MAB
Wei et al. [68]	Multi-cell	Delay	AC-based DRL
Li et al. [69]	Multi-cell	Delay	Dueling DQN
Zhang et al. [71]	Multi-cell	Cache hit rate, backhaul usage	MAB-based RL
Sengupta et al. [72]	Multi-cell	Cache hit rate, backhaul usage	MAB-based RL
Thar et al. [73], [74]	Virtualized multi-cell	Cache hit rate, backhaul usage	Q-learning for LSTM/CNN/ CRNN selection
Liu et al. [77]	Multi-cell	Cache hit rate	Dueling DDQN
Wang et al. [81]	Multi-cell	Cache hit rate	DDQN-FL
Guo et al. [82]	Multi-cell	Cache miss number	DRL
Garg et al. [83]	Multi-cell	Success probability, cache refresh rate	Q-learning
Fang et al. [84]	Multi-cell	Equivalent throughput	DQL
Cheng et al. [86]	Multi-cell	System throughput	Bayesian learning and RL
Garg et al. [87]	Multi-cell	Success rate	Q-learning
Qian et al. [88]	Multi-cell	Bandwidth fluctuation	Hierarchical RL
He et al. [89]	Multi-cell SDN	Spectral efficiency, backhaul usage	DRL
Chou et al. [90]	Multi-cell	QoE	DDPG-based DRL
Luo et al. [93]	Multi-cell SDN	QoE, energy consumption	A3C-based DRL
Wang et al. [94]	Multi-cell	QoE, cache hit rate, backhaul usage	AC and LSTM-based DRL

processing and low-complexity remote radio heads (RRHs) for wireless access. At the same time, C-RANs may stress the fronthaul due to the massive number of content requests. Thus, F-RANs have been proposed, as a promising technology towards reducing the load of the fronthaul in cellular networks. The APs in F-RANs and the overall F-RAN architecture is illustrated in Fig. 7 where F-APs are partly responsible for baseband processing, while offering edge storage and computing resources. The adoption of RL-aided caching and resource management in F-RANs represents an important research area that has attracted with various contributions recently.

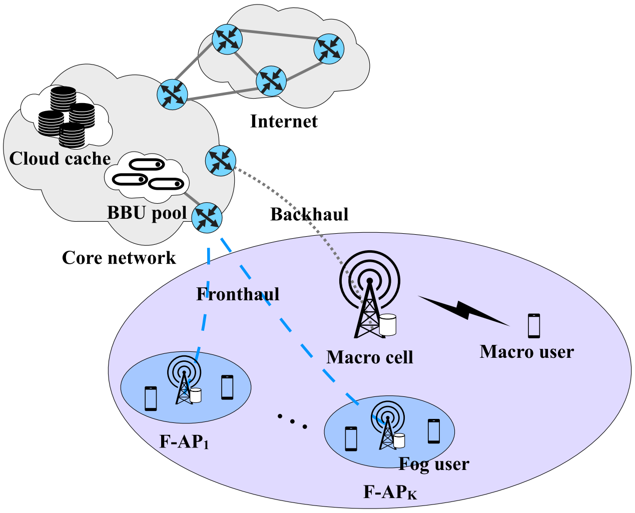

FIGURE 7. A mobile wireless network relying on the F-RAN architecture.

A. DELAY REDUCTION

In [96], a network comprising multiple cache-aided F-APs is considered, targeting the minimization of the average delay under temporal channel variations, user mobility and varying user preferences. For this purpose, the cache update content process at the F-APs is model as an MDP, while dueling DQN is employed to solve the MDP problem without knowing the state transition probabilities. Dueling DQN estimates the state value and the rewards of the different actions, facilitating cached content replacement with appropriate contents in each transmission period. Comparisons with policy-based caching, including FIFO, LRU and LFU shows that the dueling DQN increases the average cache hit rate and reduces the average transmission delay for varying storage size and number of users. Meanwhile, important performance gains are harvested when a joint radio resource management (RRM) and cache update policy is developed.

Another work targeting latency minimization in F-RANs was presented in [97]. The joint optimization of proactive caching and power allocation in the F-RAN downlink with multiple APs and a DRL controller at the centralized cloud is studied. It is shown that optimizing the latency while considering user QoS, storage and transmit power resources results in a non-convex mixed-integer nonlinear fractional programming (MINLFP) problem. So, latency minimization is modeled as an MDP without a priori knowledge of the state transition probabilities and a DQN-based algorithm derives the optimal solution. DQN is implemented using TensorFlow, simulating a network comprising 10 F-APs and 5 RRHs with 30 users. Comparisons with benchmarks, relying on weighted minimum mean square error, Q-learning, fixed, and random resource allocation reveal that DQN achieves improved convergence while latency is reduced by 18% to 49% compared to the other schemes. Meanwhile, its cache hit rate is higher than that of LFU, LRU and FIFO at 89%, while LFU provides 81% hit rate and the other two policies provide 75% hit rate.

In an F-RAN, comprising a cloud-based BBU and multiple cache-aided enhanced RRHs (eRRHs), the authors in [98] target the delivery latency minimization over X-haul links when the file popularity is time-varying and unknown. The proposed model-free RL-based scheme relies on linear VFA, adaptively activating the backhaul or the fronthaul at each transmission period. Backhaul activation updates the cache content at the eRRHs, reducing the latency at future transmission periods, while fronthaul activation leads to cooperative transmissions, reducing the latency at the current transmission period. Performance evaluation when the BBU is located at the cell center while eRRHs and users are circularly placed shows that for small eRRH cache sizes, i.e., less or equal to 4 files, fronthaul selection guarantees lower latency, due to the limited caching. On the contrary, for cache sizes larger than 4 files, backhaul activation is superior. Overall, RL provides the lowest latency compared to other schemes, relying on only fronthaul/backhaul selection, greedy fronthaul/backhaul selection and offline caching of the most popular files.

Further service delay reduction results were given in [99] where the joint optimization of content caching, computation offloading, and radio resource allocation in fog-enabled IoT is studied. AC-based learning is adopted, relying on DNN for Q-value VFA of the critic, while the actor policy is represented by another DNN. Also, RL divergence is mitigated by employing fixed target network and experience replay. At the same time, the Natural policy-gradient method is used, being more efficient than Standard policy-gradient and guaranteeing that convergence to the local maximum is avoided [100]. Results for a library of 1000 different contents highlight that the proposed DRL solutions offers reduced service delay for cache sizes below 500 contents, still outperforming conventional content popularity-based caching for cache sizes larger than 500 and up to 1000 contents where identical performance is observed.

In [101], the stress on the fronthaul link is alleviated and transmission delay is reduced through a cooperative DRL-based strategy. More specifically, cooperative caching among the F-APs is employed. Initially, content popularity prediction is performed by relying on the topic model which identifies popularity growth patterns. As a multi-variable reward function is formulated, DRL content caching strategy is used to determine the content placement policy by exploiting the content popularity prediction results. As observed in the performance evaluation results, DRL reduces the average transmission delay compared to policy-based algorithms.

B. CACHING EFFICIENCY

Cache hit rate improvement in F-RANs is the main topic of [102]. Here, a distributed edge caching scheme is developed, relying on Q-learning with VFA for reduced complexity and improved convergence. Since unknown content popularity is considered, a content request model based on hidden Markov process is proposed to identify the characteristics of the varying spatio-temporal traffic requests. At the same time, the distributed learning scheme enables each F-AP to independently determine the optimal caching policy, thus avoiding network coordination overheads. Performance evaluation in a network with 20 F-APs, each having a fixed cache size of 5 files, shows that Q-VFA-learning better adapts to content popularity fluctuations and dynamic user arrivals and departures from the network, compared to LRU, LFU and Q-learning without VFA.

Distributed edge caching with dynamic content recommendation is the topic of [103]. The authors aim at determining an efficient joint caching and content recommendation policy to reduce the cost of cache replacement in F-APs when user requests datasets are not available. Thus, a per-user request model is presented to characterize the fluctuation of requests after content recommendation. Next, a DDQN-based caching algorithm is formulated to reduce the large state and action spaces and guarantee faster convergence. Simulations reveal that the DDQN-based policy offers the highest net profit compared to LRU, LFU, Q-learning and DQN alternatives, for different cache sizes.

C. SPECTRAL EFFICIENCY

The improvement of QoS given the fluctuations of user preferences is investigated in [104]. In order to better exploit the caching resources of F-APs, random linear network coding (RLNC) is used to divide the files into subfiles and distribute them across the F-APs. By exploiting the accumulated user requests and considering the successful transmission probability as the reward for the operation of DRL, the optimal caching strategy is derived. Simulations, using TensorFlow reveal that the integration of RLNC into the proposed learning solution can save significant caching resources and increase the successful transmission probability in F-RANs, compared to uncoded caching.

The reduction of the fronthaul load is the main target of [105]. In order to improve the content placement process with unknown file popularity, a two-phase procedure is proposed. In the first phase, unsupervised learning-based feature extraction is employed to extract the content popularity from the frequently collected user requests. In the second phase, DRL with transfer learning is adopted and exploits the predicted file popularity of the previous phase to determine the optimal content placement strategy.

Performance comparisons with traditional neural network-based algorithms indicate that the unsupervised learning-based popularity prediction scheme improves the prediction accuracy, independently of the time-slot duration, while DRL with transfer learning outperforms both LRU and LFU, in terms of fronthaul load reduction.

In [106], the joint optimization of caching and radio resources is targeted. Following a hierarchical approach, a cloud-based cache resource manager aims at maximizing the system throughput and minimizing the storage cost at the F-APs, in the long-term. Meanwhile, F-APs are responsible for RRM in the short-term, considering content placement, channel state information (CSI) and user requests. Moreover, interference mitigation is guaranteed by enabling the F-APs to form clusters and perform joint transmissions to the users. In order to achieve improved performance, multi-agent RL is employed, creating one agent per each file and F-AP pair, jointly learning the caching strategy by utilizing historical CSI and user requests data provided by the network information server. Performance evaluation shows that the efficiency of the multi-agent RL-based resource management surpasses that of other schemes, based on full caching, no caching and fixed probability caching.

Joint user association and content placement for network payoff maximization is the topic of [107], in a two-tier network, consisting of a massive MIMO macro BS and a group of F-APs. Here, network payoff is defined as the ergodic rate performance utility minus the fronthaul cost for cache replacement. In this setting, game theory is invoked, formulating a hierarchical Stackelberg game where at short time scales, users act as followers, dynamically adjusting their F-AP selection, according to content placement status, while at long time scales, the F-APs act as leaders, updating their caches, based on the user association status and the content popularity prediction of a central unit, located at the core network. Towards providing low-complexity and accurate popularity prediction, a stacked auto-encoder (SAE)-based scheme is adopted. Regarding content placement, a DRL-based algorithm, extending [54] is developed. The DRL architecture is based on online DQN learning where a greedy algorithm selects an action from the state space and offline DNN and replay memory creation, executing specific optimization and storing historical information. Performance evaluation shows that DRL offers an average prediction accuracy of 90%, while baseline DNN- and CNN-based algorithms achieve 80% and 70% accuracy, respectively. Finally, compared to LRU and LFU, DRL yields the highest reward, better capturing the effect of user requests and the amount of data routed through the fronthaul.

Table 4 summarizes the works on RL-aided caching in F-RANs, their objectives and corresponding RL techniques.

Lessons learned: F-RANs must maintain low fronthaul load in order to avoid excessive delays. Thus, DRL with transfer learning provides fronthaul load reduction and adaptive fronthaul/backhaul activation for content placement guarantees low end-to-end delay. As RRHs might be resource-constrained, transfer learning can complement RL-aided solutions towards high edge caching efficiency. In general, most works on F-RANs develop joint RL-aided resource allocation and content update schemes, outperforming policy-based caching. Hybrid schemes offer high offloading efficiency through unsupervised learning-based popularity prediction and DRL-based content placement [107]. Also, distributed learning employs F-APs to independently determine the optimal caching policy, thus reducing network coordination overheads. However, RL-aided edge caching for improving the energy efficiency of F-RANs has not been extensively investigated, while mobility-aware algorithms are mainly studied in [96] and a gap exists regarding highly mobile F-RAN environments. Finally, RL-aided proactive content pushing and recommendation that can mitigate network load variation, especially in the fronthaul of dense F-RAN settings have not been studied.

TABLE 4. List of works focusing on reinforcement learning-aided caching for Fog RANs.

Reference	Performance target	RL solution
Guo et al. [96]	Delay	Dueling DQN
Rahman et al. [97]	Delay	DQN
Moon et al. [98]	Delay	Model-free RL
Wei et al. [99]	Delay	AC-based DRL
Jiang et al. [101]	Delay	DQN
Lu et al. [102]	Cache hit rate	Q-VFA-learning
Yan et al. [103]	Cache replacement cost	DDQN
Zhou et al. [104]	Average success rate	DRL
Zhou et al. [105]	Fronthaul usage	DRL with transfer learning
Sun et al. [106]	System throughput, storage costs	Multi-agent RL
Yan et al. [107]	Ergodic rate, fronthaul usage	DRL

VI. COOPERATIVE NETWORKS

Cooperation among network nodes has been considered as a viable means for enhancing the quality of communication by improving the wireless conditions through increased diversity and intelligent transmission scheduling [37], [108], [109]. Furthermore, data buffering at edge nodes, in the form of dedicated relays or user devices provides reduced outages and higher data rates [38], [39], [110], [111]. In the context of edge caching, cooperative schemes can be applied both in content caching in a distributed manner, as well as by employing user devices to cache content of other users.

A. COOPERATIVE CACHING

A general network architecture where cache-aided BSs cooperate and share data through X2/Xn interface in order to avoid constant data fetching from the core network is depicted in Fig. 8.

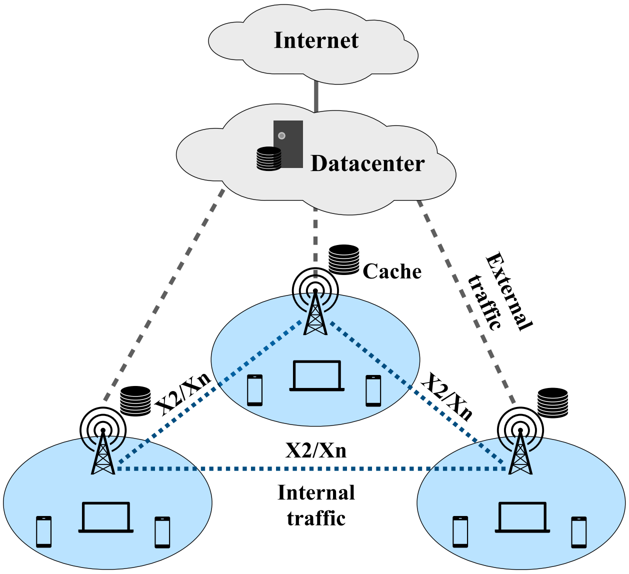

FIGURE 8. A network of cooperating BSs where data from the local cache is shared through the X2/Xn interface and constant data fetching from the core network is avoided.

1) DELAY REDUCTION

In [112], the authors solve the problem of content caching, using multi-agent AC DRL in which edge nodes adaptively learn their caching policies. More specifically, in dynamic environments, CognitiveCache is proposed, enabling edge nodes to learn their best caching policies and collaborate with their neighboring nodes to optimize content placement, thus reducing latency and transmission cost. RL tackles volatile and unreliable environments where there is no global knowledge. Single-agent DRL-aided caching has been proposed in [113], [114] where a single edge node makes suitable caching decisions. This process requires from every single node to have its own caching policy and a single central agent to make the global decisions resulting in a huge action space [94]. CognitiveCache offers better convergence than [94] while comparisons with DQNCache [114], Prob-Cache [115], LRU and LFU show that it reduces latency by 33%, 47%, 66%, 71% and transmission cost by 23%, 75%, 83% and 87%, respectively.

A slightly different problem is considered in [116] where the capability of users to offload computing tasks to edge computing nodes is examined. In this context, the coordination between edge computing nodes for the management of the compute and cache resources is investigated. Several challenges have to be addressed, such as the uncertainty of the computing task, the workload scheduling of a single node but also, the resource allocations of multiple nodes during the computation of a collaborative task. Last but not least, low latency of collaborative computing has to be ensured. A simulation environment has been developed in Python and DDQN is compared against dueling DQN, DQN and Natural Q-learning, in terms of task computation failures, revealing better performance, due to improved caching decisions, resulting in reduced delay.

Single and joint transmission of nodes are considered in [117] where storage- and transmission-level cooperation is exploited to optimize content caching and updating for video delivery. The authors formulate the problem, as an MDP where the reward is mapped to the level of delay reduction. They develop an online RL algorithm to search for the optimal caching policy, updating cache contents in an online manner. In addition, the proposed Q-learning algorithm is extended with linear approximation, thus facilitating its application in settings with a large number of contents. Comparisons in terms of normalized delivery delay are given against two conventional optimization algorithms. The first algorithm has been presented in [118] and allocates part of the caches in each cluster to store the most popular content in every edge BS, while the remaining parts cooperatively cache different partitions of the less popular content at different nodes. The second algorithm is the FemtoCaching strategy of [119], employing nodes with low-rate backhaul capacity but large storage to cache popular video content while non-cached files are transmitted by the cellular BS. Results suggest that the proposed strategy provides a delay reduction of at least 6% and 15%, compared to the first and second algorithm, respectively.

One of the main DRL challenges is that its agent must observe enough environmental features to ensure decision accuracy. The authors of [9] propose AC-based DRL for multi-cell and single-cell cooperative networks where BSs compete with each other for wireless access and also cooperate towards delay reduction. Here, the agents decide their individual caching actions while cooperating with each other, resulting in a centralized critic network and a decentralized actor network. In this framework, the agents update the actor network with their observations and the critic network with the complete state space. Compared to LRU, LFU and FIFO in a scenario with time-varying content popularity, the AC-based DRL offers the best long-term performance and each time the popularity distribution changes, it is able to converge to the previous delay performance level.

The decentralized cooperative caching towards content access latency minimization is the topic of [120]. The proposed solution relies on FL and DRL and uses two training rounds. During the first round, the BSs learn a shared predictive model, using training parameters, as the initial input of local training. In the second round, the BSs upload the near-optimal local parameters, as input to the global training. The proposed FL and DRL solution is compared against LRU, LFU, FIFO, achieving improved performance, while the decentralized cooperative approach performs very close to an alternative centralized DRL algorithm.

The authors of [121], [122] propose a MAB-based cooperative caching policy to reduce the download latency in a multi-cell MEC network. The difference of [122] to [121] is that, while the user's preference is unknown, the historical content demands are available. In contrast to other algorithms, assuming knowledge of the content popularity distribution [123], this work does not depend on previous knowledge of content popularity and user preference. A Q-learning algorithm operates on the MEC servers, which they train with their local caching decisions and subsequently, they combine the results with the decisions of other MEC servers. Also, a combinatorial MAB upper confidence bound method is employed to reduce the overall complexity. Performance comparisons of the MAB-based algorithm are conducted against a single-agent RL caching algorithm, a modified version of LRU and a randomized replacement caching algorithm. Two different experiments are performed, focusing first on the cumulative caching reward and the cumulative number of cache hits over time and second on the weighted average downloading latency and the cache hit rate when the storage capacity of each server varies from 10 to 100 units. Results suggest that the weighted average download latency can be reduced by 8%, 21% and 24%, respectively, while the cumulative number of cache hits is higher by 41%, 157% and 246%, respectively.

The joint investigation of cooperative edge caching at BSs and request routing, towards reducing the content access delay of the users and improving their QoS is at the epicenter of [124]. This problem is modeled, as an MDP and DDQN-based learning is adopted for providing QoS guarantees and backhaul offloading without statistical knowledge of the content popularity. As a reward, this scheme considers the long-term average content fetching delay of the end-users. Trace-driven simulations suggest that DDQN-based learning offers a performance gain of 7%, 11% and 9%, in terms of delay reduction when compared to LRU, LFU and FIFO caching schemes, respectively. At the same time, the performance gap against an oracle algorithm, having a priori knowledge of the users' preferences and behavior is at 4%.

2) CACHING EFFICIENCY

The authors of [9] formulate a cache hit rate maximization problem and solve it through AC-based DRL in cooperative topologies. Performance comparisons against LRU, LFU and FIFO for different cache sizes reveal that by employing a centralized critic network, the AC-based DRL learns the impact of individual decisions on the overall cache hit rate, striking a balance between the cache hit rate of each agent and that of the overall system. As a result, the cache hit rate of the learning-based framework surpasses the performance of the three policy-based caching schemes.

Then, in [125], SDN and C-RAN architectures are combined for improved cooperative edge caching. The BBUs of C-RAN are equipped MEC capabilities, forming intelligent BBU pools, performing signal processing and data pre-processing and improving the performance of AI applications. In order to maximize the cache hit rate and the cache capacity usage, DRL is employed, considering both global and local cache information. The proposed Q-learning-based collaborative cache algorithm (QLCCA) selects among three different actions, in terms of cache management. The first action is to cache the most popular data in the local cache of BBUs. The second action exploits neighboring BBUs with short transmission delays, coordinated by the SDN controller and the available data at the global cache, formed by the caches of multiple BBUs. As a third action, QLCCA adopts random data caching. A roulette-based policy is presented to optimize action selection, applying weights to each one. Performance comparisons, using Matlab˙ shows improved cache hit rate, as the number of content types increases, over alternative local and global caching schemes.

Wireless content delivery and replacement are the focus of [126]. Multi-agent Q-learning is used, modeling the problem as an MDP, where each small BS is considered as an agent. The caching policy aims at maximizing the cache hit rate and combines LFU and LRU policies, achieving better performance than the standalone versions. In the simulations, the shot noise model is adopted [127] for determining the content request pattern over time, generating content requests with temporal correlation. Results show that under temporal correlation, the learning-based approach reaches a hit rate of 79%, compared to 77% and 76% for LFU and LRU, respectively. Meanwhile, delay performance is improved and the multi-agent Q-learning guarantees a delay of 1.1 time-slots, while LFU and LRU provide delays of 1.3 and 1.45 time-slots, respectively. This study is highly related to [128] which also models content replacement as an MDP. However, the two works aim at different objectives, since in [126] the goal is to maximize the cache hit rate while the objective of [128] is to minimize the system transmission cost. Moreover, the work in [126] adopts more realistic simulation parameters and network procedures, as multiple contents can be simultaneously replaced.

The authors in [129] aim to solve the cooperative content sharing problem towards reducing the content transmission cost. A partially observable MDP formulation is adopted and solved by using multi-agent AC DRL, optimizing the content fetching cost from the local BS, neighboring BSs and remote servers. First, a communication module is proposed to acquire and share the action variability and observations of the BSs. Then, the statistics of content popularity at each BS are obtained through a variational recurrent neural network and subsequently, AC DRL-based cooperative edge caching is employed, enabling the BSs to cooperate, and leverage the interdependency on the local and global states of the distributed decision making of each agent. Performance evaluation in a multi-cell network reveals that multi-agent AC DRL improves the caching efficiency compared to multi-agent AC, DRL and LRU.

3) SPECTRAL EFFICIENCY

In [130], maximum distance separable codes are used for cooperative coded caching at small BSs in a scenario with time-varying and unknown content popularity. Aiming to address this challenge and guarantee reduced fronthaul load, a multi-agent DRL solution is presented for cache update. Initially, the dynamic coded caching is modeled, as a cooperative multi-agent MDP and due to maximum distance separable coding, the cache update decision is characterized as a constrained RL problem with continuous decision variables. So, DDPG with homotopy optimization is presented, developing a continuous transformation of the original problem and exploiting its solution for tackling the original problem. Simulations show that homotopy DDPG surpasses the performance of standalone DDPG under different network control methods and overall, the decentralized design outperforms its centralized counterpart.

B. DEVICE-TO-DEVICE NETWORKS

The massive connectivity requirements necessitate communication paradigms, deviating from conventional architectures. Thus, D2D communication has been proposed, as a remedy for excessive cellular traffic, enabling users to directly exchange or relay data for cell edge users. Decentralized caching at the users devices and intelligent D2D resource allocation can provide several gains to wireless networks, minimizing energy consumption, delay and backhaul usage [131]–[134].

1) ENERGY EFFICIENCY

The authors in [135] focus on improving the caching efficiency in D2D networks and minimizing the energy cost by pre-fetching files at user devices and small BSs. D2D communication offloads part of the cellular traffic, exploiting the caches of users and their spatial distribution. Content request behaviour is modeled as an MDP and RL is applied to discover the file popularity and user preference distributions. Because of the different capabilities and algorithmic complexities, Q-learning is applied on users' devices and DQN on small BSs. Comparisons against optimal caching with known popularity, random file caching and no user caching, depict that the proposed learning-based algorithm closely follows the performance of optimal caching, in terms of energy consumption and cache hit rate, independently of the number of files and user preferences profiles. More specifically, its cache hit rate is around 85% while the hit rate of the random caching scheme is 64.5%.

The work in [136] focuses on content placement and delivery strategies in cache-enabled D2D networks, aiming at minimizing the content delivery delay and the power consumption. ESN-based learning is employed for predicting the content popularity and user mobility patterns, determining what and where to cache. Then, content delivery is optimized by relying on a DQN-based algorithm, exploiting CSI and content transmission delay observations to decide which actions should be taken. The DQN-based caching strategy is evaluated in a multi-user network and compared against Q-learning and random caching. It is observed that due to the larger action-state space, the reward in the DQN case is higher than that of Q-learning, while random caching provides significantly smaller rewards.

2) DELAY REDUCTION

Apart from [136], jointly tackling energy and delay performance concerns through two-step learning, i.e., ESN-based prediction and DQN-based content delivery, there have been various RL strategies focusing on delay reduction.

Learning-based caching strategies are proposed in [137] for D2D caching where multi-agent MAB modeling is adopted. Since the action space is too large, there is no instantaneous knowledge of the content popularity profile. More specifically, the users follow a Q-learning approach where each one learns the Q-values through their own actions and the actions of the other users. In order to reduce the action space and the overall complexity, a belief-based modified combinatorial UCB approach is adopted for regret minimization, in terms of download latency. Comparisons include random replacement, LRU and LFU. More specifically, two scenarios were examined, when the size of the cache increases and when the number of users increases. In the first scenario, the gain of the proposed algorithm in terms of ADL is 13% to 24% and in terms of cache hit rate, 22% to 194% which also depends on the number files. In the second scenario, the gain, in terms of cache hit rate ranges between 35% to 123%.

The paper in [138] focuses on delay reduction with D2D-aided caching where content popularity and user location statistics are time-varying. In this setting, a stochastic game is formulated to derive a cooperative cache placement policy. Moreover, the user reward to participate in the caching process is designed as the difference among the caching incentive and the transmission power cost. Towards solving this stochastic game, a multi-agent cooperative alternating Q-learning algorithm is employed, alternatively updating the user cache placement policy, depending on the stable policy of other users, until all users obtain a stable cache placement policy. Simulations show that lower delay is achieved, while the backhaul load of the cellular network is reduced.

3) SPECTRAL EFFICIENCY

The authors in [139], [140] address D2D mobile edge caching for increased offloading. A content delivery market formulation is given and blockchains and smart contracts are employed. Here, edge caching comprises a content placement and cache sharing problem, as well as the verification that the caching actions of the peers are recorded and handled in a trustworthy manner. In this topology, there exist different subsystems, performing necessary procedures, integrated by a cache and blockchain controller. First, the caching subsystem associates the peer contributions with their willingness for D2D data sharing. The blockchain subsystem performs transaction verification at low cost and latency and ensures system scalability. Both problems are formulated as an MDP and are addressed by a DQN algorithm. Performance evaluation compares DQN against DQL [141], [142] and a greedy scheme, employing each node to cache the most popular content within its coverage. As

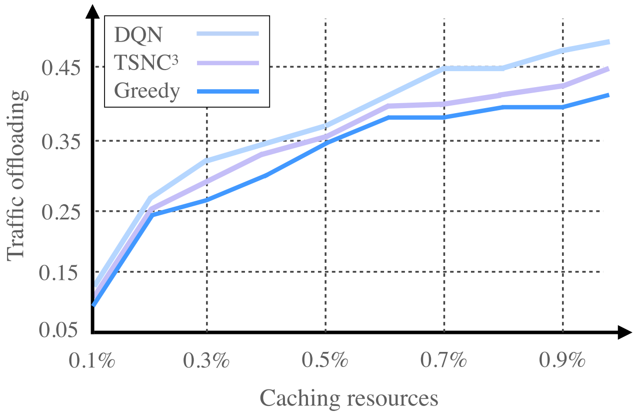

FIGURE 9. Traffic offloading versus caching resources [140].

shown in Fig. 9, improved offloading is provided by DQN for increasing caching resources at each user device, highlighting the importance of D2D communication and the necessity for incentives to increase user participation.

Mobile social networks are examined in [141]–[143], focusing on the impact of social relationships. A DRL approach is presented where each agent receives measurements and observations from its peers. This information is combined with the wireless channel conditions and the trust value of every peer is given as input to a DNN which outputs the proposed actions. Here, the reward is mapped to the increase of MNO profits through improved backhaul usage for video content delivery. These reward observations are also used to train and optimise the neural network. Major features of these works include the social trust scheme, the use of Bayesian inference and the Dempster-Shafer theory for the evaluation of direct and indirect observations [144]. Simulations are presented using TensorFlow, as well as comparisons with a scheme without indirect observation, a scheme without edge computing [145] and a scheme without D2D communications [146]. Also, it is observed that DRL increases the backhaul usage, benefiting the MNO profits, independently of the number of content types, while the total utility performance is better than that of the other schemes for different numbers of malicious D2D transmitters.

In D2D-aided information-centric wireless networks, collaborative caching at the user devices can result in improved spectral efficiency and offloading. Thus, the authors in [147] study resource allocation and power control in a small-cell MEC network with D2D communication. Initially, depending on whether or not the user cache is not empty, a selection among cellular or D2D communication is performed. When D2D communication occurs, channel reuse increases and an efficient power allocation scheme is devised. A policy-gradient DRL approach is presented to select the power levels, using the Gaussian distribution, as well as softmax channel selection for spectral efficiency maximization and interference minimization. Comparisons with DQN reveal improved spectral efficiency and reduced interference, since policy-gradient DRL selects a power level from a continuous state space, while DQN selects among discrete values.

4) CACHING EFFICIENCY

The work in [148] identified two challenges that need to be addressed in D2D networks. First, the need for caching decisions towards maximizing the probability that the requested content will be cached in a neighboring node given the storage constraints and the plethora of the available content. Second, the level of offloading when multiple helper nodes are able to cache and deliver the desired content to another user. To address these issues, a caching strategy considering parameters, such as the predicted content popularity, user preferences, user activity level, and social relationships is proposed. For characterizing the offloading potential of users, the expected correlation coefficient is introduced, capturing the offloading probability and the offloading gain after a D2D content delivery event. At the same time, D2D pairs by a combinatorial MAB-based online learning algorithm,. Comparisons with random caching, the least frequently caching and matching algorithm of [149] and the collaborative caching and auction matching algorithm of [150] are presented, using real-world traces from the Unical/SocialBlueconn dataset to model the D2D topology. Results reveal that the MAB-based algorithm achieves a 53% offloading ratio, contrary to 45% by collaborative caching and auction matching, 22% by least frequently caching and matching and 10% by random caching.

Table 5 summarizes the studies on RL-aided caching solutions in cooperative networks.

Lessons learned: Compared to non-cooperative multi-cell networks, data sharing among edge nodes increases the degrees of freedom when designing RL-aided caching strategies, leading to improved performance. In cooperative caching multi-agent learning, edge nodes determine their caching policies and collaborate with neighboring nodes to optimize content placement at a system level. A popular approach is to develop AC-based DRL schemes where the agents at the edge nodes update the actor network with their observations and the critic network with the complete state space through cooperation. Cooperative architectures can better support distributed edge caching strategies when determining the cache resource allocation, caching the most popular content in every edge BS and different partitions of less popular content at different nodes. Relevant studies have been shown to improve the cache hit rate performance and reduce the communication overheads [118].

In D2D edge caching networks, on the one hand, the caches of the users and their spatial distribution can be used to improve the caching efficiency. However, user devices are computationally and energy constrained and simpler learning algorithms should be used at a local level, while keeping more complex DRL solutions for infrastructure-based nodes. Moreover, trust and privacy are major concerns and solutions include the use of smart contracts and blockchains, verifying

TABLE 5. List of works focusing on reinforcement learning-aided caching for cooperative networks.

Reference	Performance target	RL solution
Radenkovic et al. [112]	Delay, transmission cost	AC-based DRL
Ren et al. [116]	Delay	DDQN
Lin et al. [117]	Delay	MDP
Luo et al. [9]	Delay, cache hit rate	AC-based DRL
Wang et al. [120]	Delay	FL and DRL
Jiang et al. [121], [122]	Delay	MAB-based RL
Li et al. [124]	Delay	DDQN
Chien et al. [125]	Cache hit rate	Q-learning
Sung et al. [126]	Cache hit rate	Multi-agent Q-learning
Gu et al. [128]	Transmission cost	Q-learning
Chen et al. [129]	Transmission cost	Multi-agent AC-based DRL
Wu et al. [130]	Fronthaul usage	Homotopy DDPG
Tang et al. [135]	Energy consumption	Q-learning and DQN
Yin et al. [136]	Energy consumption, delay	DQN
Jiang et al. [137]	Delay	Combinatorial MAB-based RL
Zhang et al. [138]	Delay	Multi-agent Q-learning
Zhang et al. [139], [140]	Traffic offloading	DQN
He et al. [141]–[143]	Backhaul usage reduction	DRL
Wang et al. [147]	System throughput	Policy-gradient DRL
Sun et al. [148]	Offloading ratio	Combinatorial MAB-based RL

that the caching actions of the peers are recorded and handled in a trustworthy manner [139], [140]. Also, an additional parameter that can be exploited is related to social relationships among users, enabling peers with high trust values to share their observations, serving as input to RL-aided caching schemes [141]–[143].

VII. VEHICULAR NETWORKS

RL-aided edge caching in vehicular networks must address various challenges, related to the dynamicity of the network, the computing and power resources of vehicles and RSUs, as well as the vast amount of mobile data. Different scenarios of vehicular communications, such as V2V or vehicle-to-infrastructure (V2I), either towards a macro BS or an RSU are shown in Fig. 10. There, RL agents located at vehicular and infrastructure-based nodes assume the task of deciding the caching strategy under specific performance targets. For example, video content related to in-car entertainment can be cached at a nearby RSU while road safety and traffic data might be cached at a nearby vehicle, thus achieving low latency. The works presented in this section cover various aspects of vehicular networks, aiming at energy efficiency, delay reduction and improved caching performance.

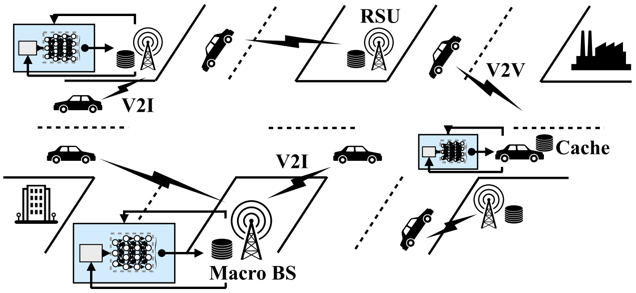

FIGURE 10. Various cases of vehicular edge caching relying on RL-aided nodes.

A. ENERGY EFFICIENCY

The paper in [151] examines the difficulties that edge servers experience due to the heterogeneity of on-vehicle applications and the time variability of content popularity. A learning framework is proposed, allowing cross-layer offloading and cooperative multi-point caching. In cross-layer offloading, a computationally heavy task can be offloaded to the next computation layer, such as RSUs and the latter can subsequently offload it to BSs. Thus, a DDPG RL-based resource allocation scheme is proposed, maximizing the system utility which considers the energy consumption, and the computation and caching efficiency. Simulations demonstrate the effectiveness of the proposed scheme and reveal a trade-off among using edge servers to store the past experience of user preference for improved prediction accuracy or using these resources for content caching to improve system utility.

The study of [152], [153] solves a two-fold problem. First, establishing a network for secure content caching and then, ensuring efficient content caching in a volatile network. To address the first problem, a blockchain enabled distributed content caching network is proposed, allowing the BSs and the vehicles to establish a secure peer-to-peer transaction environment. BSs maintain the permissioned blockchain and the vehicles perform content caching. To tackle the second problem, DRL-based content caching is presented, taking mobility into account while block verification is accelerated. The reward function considers the total consumed energy for content transmission and content caching. Also, a security analysis is conducted, showing that blockchain content caching provides security and privacy protection with low-energy consumption. Performance analysis using a real dataset shows that the proposed algorithm achieves a 86% of successful content caching requests against 76% of a greedy algorithm and 5% of a random content caching algorithm.

B. DELAY REDUCTION

The topic of [154] is content delivery delay minimization, presenting a framework where RSUs collaborate with vehicles to cache popular contents. Multiple communication scenarios are investigated, such as V2V links, vehicle-to-RSUs and vehicle-to-macro BSs. In this context, a DDPG DRL-based solution is employed to solve the content edge caching and delivery problem. Comparisons against random edge caching and delivery, and DDPG without bandwidth optimization are given. The proposed algorithm demonstrated significant improvements against the two benchmarks, in terms of content delivery delay and cumulative total reward, while exhibiting faster convergence.

The authors of [155] exploit the trilateral collaboration among vehicles, macro BSs and RSUs for low-latency communication in vehicular edge networks, optimizing the content placement and delivery. The problem is modeled as a double time scale (DTS) MDP, considering that network changes are more frequent than content changes, in time. Content placement considers content popularity, vehicle path and resource availability, as the conditions to optimize in the large time-scale, while in the small time-scale, vehicle scheduling and bandwidth allocation are performed for content delivery latency minimization. So, a DDPG framework is adopted for obtaining a sub-optimal solution with low computational complexity. Performance evaluation is presented, taking into account content delivery latency, cache hit rate and system cost where substantial improvements against random caching and non-cooperative caching (e.g. 135.62% and 34.33% hit rate increase, respectively) are shown.

The authors of [156] elaborate on vehicular edge computing, using RSUs for computation offloading. The limited spectral resources to support the communication and computation offloading in edge computing and autonomous driving pose significant challenges. So, a unified model for data scheduling is developed, allowing V2V and vehicle-to-RSU communication and collaborative computing. Targeting to improve the data processing performance, a penalty mechanism is introduced, activated when the data processing deadline is not satisfied. For this purpose, a DQN algorithm is developed to derive the collaborative data scheduling strategy, minimizing the data processing cost without increasing the delay. Comparisons with a DQN benchmark without a separate target Q-network shows that the latter's reward performance requires additional episodes and exhibits higher fluctuation. Regarding the average reward of long-term data scheduling, DQN with separate target Q-network is compared with benchmarks, employing only vehicular, RSU or non-collaborative data offloading, demonstrating processing cost reduction and delay guarantees.

C. SPECTRAL EFFICIENCY

The paper in [157] focuses on content caching, computing and networking in vehicular networks. A DRL algorithm, using DQN for Q-value approximation is presented, orchestrating networking, caching and computing resources to meet the requirements of different applications. The reward function considers the MVNO revenue, consisting of the received SNR of the wireless access link, the task computation capability and the cache state. Comparisons of the DRL solution against benchmarks without virtualization, MEC offloading and mobile edge caching in a static setting, emphasize its superiority for increasing the MVNO revenue.

Another scheme focusing on maximizing the MVNO revenue is presented in [158], proposing dynamic resource allocation in vehicular networks. Reward consists of the MVNO revenue, which is modeled as a function of the received SNR, the computation capability and the access link state. By relying on SDN and information-centric networking, dynamic orchestration of computing and communication resources for virtual wireless network optimization is targeted. The authors model the resource allocation strategy as an MDP, using VFA. The high complexity of the problem is tackled using A3C-based RL and simulations demonstrate increased reward and high convergence speed, resulting in improved MVNO revenue compared to a policy-based scheme.

The authors of [159] take advantage of DRL to orchestrate edge computing and resource allocation with the goal of maximizing the MNO revenue without degrading the QoE in V2V networks. More specifically, they design a DDPG model to optimize resource allocation and task assignment in a volatile vehicular environment with mobile edge caching servers. They conduct experiments based on real traffic data and they compare the DDPG-based solution against a non-cooperative scheme, a computation offloading scheme and an edge caching scheme without computation offloading. Results indicate that the MNO profits can be significantly larger when the proposed scheme is adopted, compared to the other benchmarks.

In [40] the goal is the maximization of a reward function comprising communication, computation and data offloading metrics, while satisfying a deadline-constrained service. In this network, both vehicles and RSUs have caching and computing capabilities. They collaborate and communicate via V2V and V2I communication to improve the cache hit rate by retrieving content via nearby vehicles or RSUs. Moreover, vehicles are able to offload tasks to neighboring vehicles, RSUs and BSs. A Q-learning algorithm with multi-timescale network is proposed for the caching placement, computing resource allocation and assessment of the sets of possible connecting RSUs and vehicles. Optimal configuration of the Q-learning algorithm is performed, validating their theoretical findings and achieving significant cost gains against a random resource allocation scheme and a scheme where caching and computing capabilities are limited to RSUs.

A similar topic with [40] is studied in [160], formulating a joint caching and computing allocation problem for cost minimization under the constraints of dynamic RSU storage capacities. Multi-time scale algorithms are developed, dictating caching placement, computing resource allocation and assessment of the sets of potentially connected RSUs, contrary to connecting RSUs and vehicles, as in [40]. The developed algorithms are based on particle swarm optimization and DQL for large and small timescale models, respectively. Numerical results show significant performance gains while using optimal parameter configurations for the proposed algorithms.

In a scenario with cache capacity constraints, the authors in [161] target at data rate maximization and present a cooperative caching strategy in a vehicular network. Their approach relies on the Hawkes process to adapt to the variability of content popularity. The Hawkes process models a sequence of arrivals over time, with each one increasing the chance of a subsequent arrival for some time after the first arrival. Also, DRL is employed for determining the cooperative content caching decision where the reward is mapped to the data rate that is achieved. They compare their approach with policy-based LFU and a static caching scheme and show that the DRL-based strategy significantly improves the throughput performance.

D. CACHING EFFICIENCY

In the context of the Internet of Vehicles, edge caching can play a major role towards improving the content request process and reducing the backhaul load of fixed BSs. As content popularity is time-varying, the work in [162] aim at improving the cache hit rate in a vehicular setting through a cooperative caching strategy with content request prediction. For this purpose, a three-step process is proposed in which initially, K-means is employed to cluster the vehicles and facilitate content request and dissemination. Content request prediction is performed through LSTM, leveraging the historical content request data. In the third step, RL is used to determine the optimal caching decision which maximizes the cache hit rate. However, they have not focused in detail on the mobility of vehicles on the caching decision. Performance comparisons show that the three-step solution surpasses the content acquisition delay performance of LFU and LRU by 8% to 10%, respectively for different content popularity Zipf parameter values. Also, it achieves a cache hit rate improvement of 5% and 7% over LFU and LRU, respectively when the Zipf parameter is set to 0.7.

The integration of RL with multi-level FL is presented in [163]. Since most caching policies incur high overheads while trying to adapt to the dynamicity of content popularity and vehicular networking, a cooperative caching solution with multi-level federated RL is proposed, determining the cache update policy and content request service. The authors perform a two level aggregation to speed up the convergence rate, a low-level aggregation at the RSUs and a high-level aggregation at a global aggregator. The low-level aggregation is based on a DLR model which is trained by on-board units and provides feedback to the RSUs. High level-aggregation aims towards federated training and improves convergence. The proposed algorithm is compared against LRU, LFU and FIFO and a baseline FL DRL algorithm. The proposed algorithm has improved hit rate (10% to 15% improvement) and latency performance (24% to 28% improvement) over LRU, LFU. Moreover it converges faster than the FL algorithm without two level aggregation for different cache capacities and content volume.

Table 6 includes relevant studies on RL-aided caching solutions in vehicular networks and the performance goal they pursue.

TABLE 6. List of works focusing on reinforcement learning-aided caching for vehicular networks.

Reference	Performance target	RL solution
Dai et al. [151]	Energy consumption	DDPG-based
Dai et al. [152], [153]	Energy consumption	DRL and permissioned blockchain
Dai et al. [154]	Delay	DDPG-based
Qiao et al. [155]	Delay	DDPG-based
Luo et al. [156]	Computation delay	DQN
He et al. [157]	Backhaul usage, MVNO revenue	DQN
Chen et al. [158]	Backhaul usage, MVNO revenue	A3C-based
Ning et al. [159]	MNO revenue	DRL
Tan et al. [40]	Spectral efficiency, storage cost	DQL with multi-timescale network
Xing et al. [161]	Spectral efficiency	DRL
Wang et al. [162]	Cache hit rate	K-means and LSTM-aided DRL
Zhao et al. [163]	Cache hit rate	Multi-level federated RL

Lessons learned: Resource-demanding applications with stringent latency and reliability requirements, such as in-car entertainment, autonomous driving and live traffic monitoring pose significant difficulties to communication networks. Vehicular communication, edge caching and computation offloading from connected and automated vehicles provide important tools for satisfying such services. However, the highly dynamic vehicular environment makes the resource allocation a non-convex optimization problem with complicated objective function and constraints. Towards this end, most works develop RL-based solutions with joint communication and computation resource allocation for data and task offloading. RL has shown its potential in abstracting the parameters of vehicular networks, providing optimal resource allocation strategies, outperforming non-collaborative and policy-based approaches. Still, security and trust are major issues in vehicular applications and initial studies have shown that employing infrastructure-based nodes, such as BSs to maintain permissioned blockchains can provide secure V2V edge caching with privacy preservation [152], [153]. Furthermore, improved mobility prediction schemes are required in order to perform proactive content recommendation and caching at edge nodes, processes which can be more easily implemented in intelligent transportation systems with predetermined trajectories. In this context, hybrid learning paradigms where historical mobility data can be leveraged by a supervised learning algorithm in conjunction to online RL have not been developed.

VIII. UAV-AIDED NETWORKS

UAVs will play a vital role in 6G wireless networks. UAV-aided networks enable flexible network deployment, allocating resource where and when needed, as well as fast recovery after disasters and network outages. However, their dynamicity poses challenges to wireless networks, as the mobility of the aerial APs affects user association, interference levels and caching decisions. In this context, RL is expected to provide solutions for efficient edge caching by deploying UAVs at optimal locations and determining their trajectory and communication parameters.

A. DELAY REDUCTION

Focusing on cache-enabled UAVs, the paper in [164] examined proactive caching for reduced latency and backhaul load. Multi-objective optimization determines the minimum number of deployed UAVs, transmit power, UAV-user association, and cache location. In order to solve the multi-objective optimization, RL is adopted for user grouping, performing local search according to the optimal UAV deployment over each group. From the results, the efficiency of the RL method is shown, effectively minimizing the number of required UAVs, compared to the case without caching, as well as their 3-D placement in the network, resulting in improved cache placement and lower delay.

In cache-enabled UAV-aided NOMA networks, the authors in [165] turn to RL for leveraging the dynamic UAV mobility and content request variations. Initially, long-term optimization of cache placement, user scheduling and NOMA power allocation is formulated, minimizing the sum delay of ground users. Then, the optimization problem is converted into an MDP and Q-learning reaches to a near-optimal solution. Nonetheless, in large-scale networks, Q-learning fails to cope with the large MDP state and action spaces and VFA-based caching and resource allocation is proposed. Performance evaluation in a multi-cell environment focuses on a single cache- and UAV-aided cell. Content delivery delay and cache hit ratio comparisons include the two RL solutions, a greedy algorithm obtaining the optimal delivery delay of the current state, a fixed algorithm caching the popular contents of previous states, employing round robin-based scheduling and fixed power allocation and finally, random content caching and resource allocation. It is concluded that in small-scale networks, Q-learning provides a small performance gap compared to the greedy algorithm, while in large-scale networks, function approximation outperforms both random and fixed algorithms.

A UAV-aided small cell topology, providing virtual reality (VR) services with stringent delay constraints is studied in [166]. UAVs alleviate the burden of backhaul and access links by collecting the desired contents, transmitting them to the cache-aided small BSs, communicating with the VR users. The joint optimization of caching and transmission is solved by developing a DL algorithm, relying on LSM neural networks and ESNs, namely echo liquid state machine (ELSM) DRL. ELSM DRL identifies the relationship among actions, selection policy of small BSs and user reliability. Compared to conventional DRL, ELSM-based DRL offers increased prediction accuracy, using historical data while ESNs reduce the training complexity by adjusting only its output weight matrix and avoiding the calculation of the gradients of all the neurons. Comparisons include ELSM, LSM, Q-learning [167] and ESN-based learning [168]. It is shown that ELSM provides a 10% and 18.4% reliability improvement against ESN and Q-learning when 35 users exist in the network. In addition, ELSM converges 11.8% faster, compared to LSM in a network with 11 small BSs.

An LTE cloud network operating in licensed and unlicensed bands is the studied in [169], focusing on resource allocation for cache-enabled UAVs. The performance target here is to maintain queue stability, directly affecting the content transmission delay in the network. Constrained by the limited capacity of the UAV-cloud links, LSM is employed to help the UAVs perform content caching and resource management. LSM enables the cloud to efficiently learn user-centric information regarding content request distribution and to facilitate spectrum allocation by the UAVs. This method is extended in [170], where an optimization problem is formulated to maximize the number of users with stable queues. As a solution to this problem, a self-organized and decentralized algorithm is developed and LSM is employed for joint caching and resource allocation over both licensed and unlicensed bands. Performance evaluation reveals that LSM increases the number of users with stable queues in comparison to Q-learning with and without content caching.

B. ENERGY EFFICIENCY

Improved UAV-aided network operation in the Internet of Vehicles is the topic of [171]. Due to the increased mobility and dynamic content requests, content delivery becomes challenging. More specifically, vehicles request contents from the UAV in the downlink while the latter has to decide which popular contents should be cached from arriving vehicles in the uplink. As a performance metric, the maximization of the number of served vehicles over the UAV energy consumption is investigated and the joint problem of caching, UAV trajectory and resource allocation is tackled. However, randomly arriving vehicles make the use of traditional optimization prohibitive. Thus, after formulating the joint problem as an MDP, PPO-based DRL is employed to control UAV trajectory. Learning relies on rewarding the agent when a vehicle is served by the UAV, and penalizing the agent, according to the energy consumption incurred by UAV movement. Simulations consider a single cache-aided UAV cell and PPO is compared against stationary UAV, random UAV mobility, maximum speed selection for moving the UAV back-and-forth over the highway, as well as minimum energy UAV selection. It is revealed that PPO-based DRL balances the amount of traffic offloading and the energy consumption at the UAV, better adapting to content requests for different content popularity values.

Another RL-based solution for improved energy efficiency in cache-enabled UAV-aided networks is presented in [172]. Focusing on an urban scenario with mobile users, the storage and energy capabilities of the UAV are considered, towards maximizing the sum achievable throughput. Since this problem is shown to be non-convex, DRL is adopted for joint content placement and trajectory design. More specifically, energy-efficient UAV control is achieved by employing a DDQN for online trajectory design, according to real-time user mobility and avoiding the over-estimation of the value function of traditional DQN. Also, during the offline content placement stage, a link-based caching strategy is developed for cache hit rate maximization through approximation and convex optimization, leading to a trade-off among file popularity and diversity. In order to illustrate the performance of caching and DDQN trajectory design, a multi-cell simulation environment using TensorFlow is developed, showing that the selection of appropriate hyperparameters, such as learning rate can increase the performance of DDQN. As a result, throughput and energy consumption gains are harvested over static circular trajectory designs where mobile users are not optimally served.

TABLE 7. List of works focusing on reinforcement learning-aided caching for UAV-aided networks.

Reference	Performance target	RL solution
Dai et al. [164]	Delay, backhaul usage	Local search-based
Zhang et al. [165]	Delay	Q-learning and VFA
Chen et al. [166]	Delay	LSM and ESN-based
Chen et al. [169], [170]	No. of stable queue users	LSM-based
Al-Hilo et al. [171]	Spectral efficiency, energy consumption	PPO
Wu et al. [172]	Spectral efficiency, energy consumption	DDQN

Table 7 lists RL-aided caching solutions in UAV networks and the corresponding performance targets.

Lessons learned: The introduction of aerial nodes in 6G networks provides tremendous opportunities to edge caching applications, as UAVs can be dynamically deployed closer to the end users, improving communication and facilitating data offloading by proactive caching. However, in order to harvest the highest possible gains from UAV-aided operation, the complexity of multi-objective optimization, considering the communication parameters, caching location, trajectory design and the energy-constraints might be prohibitive. As a remedy, RL methods provide an alternative approach to achieve near-optimal operation, as long as the issue of large action and state space is addressed, e.g. through VFA [165]. Other works have developed joint content placement and trajectory design solutions [171], [172] through DRL with promising performance. However, currently, hybrid learning-based solutions, combining offline training using historical data and online RL-aided operation for faster convergence and UAV deployment are missing. Another gap that has been observed is RL-aided decentralized caching in settings where UAV swarms are deployed and a large number of caches is available. Finally, caching strategies based on the popular AC and MAB frameworks have not been considered in UAV-aided networks.

IX. OPEN ISSUES

A. PHYSICAL-LAYER ASPECTS

In this survey, the function of caching has been studied from the aspect of content placement for improving various network objectives. However, the function of caching, in the context of buffer-aided networks is related to the optimization of physical-layer characteristics, such as diversity through appropriate transmission/reception scheduling with increased degrees of freedom. Buffer-aided relaying has shown tremendous gains in different communication scenarios, increasing the transmission reliability and mitigating interference and fading [173]–[176]. For example, buffer-aided full-duplex (FD) relays can increase the flexibility of edge caching operation, establishing end-to-end communication of users with a BS, caching their desired content. However, at another instance, the relay, having already cached this content, can deliver it under more favorable channel conditions. Thus, the development of RL-aided solutions integrating high diversity data buffering techniques, considering edge caching parameters, such as content location represents an attractive research direction.

In addition, RL has been proposed as an alternative approach to conventional optimization when the derivation of optimal communication parameters entails excessive complexity and high network coordination overheads. In mobile edge networks where different cells might overlap, the design of RL-aided caching policies should not neglect physical-layer issues. These include intra- and inter-cell interference, fast fading due to mobility from receivers and transmitters, and path-loss. RL solutions should evaluate data related to signal-to-interference plus noise ratio and jointly determine the edge caching locations, BS and user association, D2D cooperation, duplexing method, modulation order, coding rate, beamforming vectors and transmit power level. MAB-based RL-aided solutions have already shown promising performance in such physical-layer-related problems [177]–[182]. Also, in many cases, edge networks comprise energy-constrained nodes with limited capabilities, such as IoT devices and the integration of wireless powered communications with RL-aided edge caching and computing should be studied [183].

B. NON-ORTHOGONAL MULTIPLE-ACCESS

Edge caching has the potential to improve the performance of mobile networks, resulting in homogeneous QoS and better backhaul/fronthaul offloading. However, the massive number of users and devices in 6G networks calls for NOMA strategies in order to better exploit the wireless resources. Recently, the integration of NOMA in mobile edge networks has been proposed, jointly determining the task allocation, caching location and power allocation for NOMA [184]–[186]. At the same time, the benefits of NOMA in buffer-aided networks have already been shown in various works and tailor-made caching policies when users are simultaneously served on the same physical resources should be devised [187]–[190]. Still, considering the large amount of network parameters, including storage, power level, spectrum and QoS constraints, conventional optimization will often fail in deriving optimal caching policies when NOMA is employed.

C. SEMANTIC-AWARE CACHING

Traditionally, each new generation of wireless networks has been designed from an information theoretic perspective towards data rate maximization and QoS provisioning. As the research on 6G communication intensifies, a different wireless networking design principle has recently been introduced, related to the semantic aspect of data. Semantic communication radically departs from the Shannon paradigm which only focuses on correct data reception by considering the impact that correct reception has on a pre-defined goal [191]. In [192], a model for semantic-empowered and goal-oriented networking has been proposed where transmitted data conveys to the end-user information that is characterized by timeliness, usefulness and value. Moreover, the survey in [193] has discussed the importance of the semantic aspect for communication among humans and machines, such as VR/AR services where data caching at edge servers provides significant offloading and latency minimization [191]. In the context of semantic-aware edge caching, content, as well as knowledge base systems and virtual machines should be moved in proximity to end-users. So, future efforts should focus on proactive virtual machine caching, offering computation and caching gains to industrial IoT and other critical services.

D. RADICAL LEARNING PARADIGMS

RL operation for improving the performance of edge networks with a massive number of users and IoT devices should aim at avoiding complex and resource-demanding learning solutions while still exploiting the large geographical distribution and heterogeneity of edge nodes. In this context, the incorporation of transfer and federated learning in RL-aided edge networks represents a fertile research area with only a limited number of contributions [81], [105], [120]. First, transfer learning is based on initially extracting features, such as file popularity on a base network with a generalized dataset. Then, these features are used to facilitate DRL agents at the edge to converge to the optimal edge caching policy, thus minimizing the energy consumption at the edge devices. On the other hand, FL leverages the observations of multiple DRL agents at different edge nodes and trains a shared model. In addition, communication costs among edge nodes are reduced, since FL uses locally stored data, only calculating updates to the global shared model of the coordinating node.

E. SECURITY, PRIVACY AND TRUST

Mobile edge networks comprise operator-owned clouds, infrastructure-based BSs and machines, as well as user devices. Caching, apart from rate and delay improvements, has the potential to improve security in such heterogeneous wireless networks, e.g., physical-layer security (PLS) [194]. While the problem has been studied extensively in different context, ranging from cellular [195] to cooperative networks [196], cooperative and low-complexity RL solutions can be implemented on a wide range of network nodes to facilitate and enhance PLS, especially when trade-offs among latency and security arise [197]. At the same time, issues of trust are raised, especially in infrastructure-less D2D-aided edge scenarios where social-awareness can be exploited [141]–[143]. RL algorithms should take into consideration the behaviour of cooperating nodes and incur penalties when malicious behavior is observed, since caching at small BSs and more importantly, at user devices can threaten user data privacy. Furthermore, in decentralized learning paradigms, such as federated learning which better suit privacy sensitive applications, it is necessary to ensure that shared models will be based on information exchange among trustworthy peers. In this area, recent works have adopted blockchains and smart contracts, highlighting their efficiency in M2M, D2D and V2V RL-aided edge caching but still, further advancements are needed [69], [139], [140], [152], [153]. In addition, further adoption of FL can alleviate privacy concerns, as for example in F-RANs where data from IoT devices are collected and at central servers for content popularity prediction. In this case, FL-based solutions maintain data locally and IoT devices train a shared learning model for content popularity prediction purposes [198]. Finally, in the context of semantic-aware learning-based cache update and content delivery strategies, issues of data privacy and trust may arise, and building upon the FL frameworks can address such concerns, as proposed in [199].

F. COOPERATIVE CACHING EXTENSIONS

Cooperative caching is examined from various perspectives and various open issues are noted. In [112], edge nodes learn their best caching policies using a multi-agent AC DLR. However, accuracy, scalability and efficiency in HetNets can be further improved through real-time heuristics and analytics. In a different topology, the solution in [116] offloads tasks to edge computing nodes, investigating the management strategy of the compute and cache resources. In this area, further research on the use of competitive bidding and allocation priorities can enable additional gains. In [9], BSs compete for wireless access and also cooperate towards reducing the average delay. The main target is to jointly perform content caching problem along with power control and user scheduling. Recently, cooperation among space and terrestrial segments has led to the formation of integrated satellite-terrestrial networks. Currently, only few studies investigate the potential of satellites to push content to cache-aided BSs, alleviating the backhaul [200].

G. VOLATILE NETWORKING TOPOLOGIES

Vehicular and UAV-aided networks represent interesting domains of edge caching where BSs, RSUs, ground and aerial vehicles collaborate for the optimal management of caching and computation resources in a highly volatile environment. Further research can be performed regarding security, resource management and mobility prediction. More specifically, the authors of [151] take into account RSUs to provide computation offloading and spectrum planning to investigate AI algorithms for efficient handover. Proactive caching and pre-allocation of network bandwidth is also the future focus of [159]. Also, the caching and computing resources orchestration for different application types, aiming at increased energy efficiency in highly mobile networks provides another interesting future direction [157].

X. CONCLUSION

Edge caching represents a major shift in network architecture design, since content is brought closer to the users in an intelligent and proactive manner. In this way, the burden in backhaul and fronthaul is relieved and repeated requests to remote web servers are avoided. Still, the optimization of edge caching performance must take into consideration several characteristics, including mobility, resource allocation, energy and storage capabilities, as well as requirements, including rate and delay. In this context, the adoption of reinforcement learning can lead to tangible performance gains at an acceptable complexity, overcoming the limitations of traditional approaches. This survey focused on reinforcement-aided edge caching in a variety of network settings, comprising fixed access points, fog-enabled paradigms, cooperative schemes, as well as aerial and ground vehicles. The discussion of the different learning solutions revealed that the fusion of learning and edge caching can result in significant benefits, independently of the complexity of the wireless environment and surpass the performance of conventional optimization solutions, while guaranteeing service requirements in an online and autonomous fashion. Finally, several open issues in the field have been highlighted, representing important future research directions and paving the way for further innovations towards realizing the 6G communications vision.

REFERENCES

[1] R. Ali, Y. B. Zikria, A. K. Bashir, S. Garg, and H. S. Kim, "URLLC for 5G and beyond: Requirements, enabling incumbent technologies and network intelligence," *IEEE Access*, vol. 9, pp. 67064–67095, 2021.

[2] A. Ghasempour, "Internet of Things in smart grid: Architecture, applications, services, key technologies, and challenges," *Inventions*, vol. 4, no. 1, p. 22, 2019.

[3] *IMT Traffic Estimates for Years 2020 to 2030*, Int. Telecommun. Union, Geneva, Switzerland, 2015.

[4] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[5] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[6] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges, and future directions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22–28, Sep. 2016.

[7] H. Aftab, J. Shuja, W. Alasmary, and E. Alanazi, "Hybrid DBSCAN based community detection for edge caching in social media applications," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, 2021, pp. 2038–2043.

[8] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 3rd Quart., 2019.

[9] F.-L. Luo, "Deep multi-agent reinforcement learning for cooperative edge caching," in *Machine Learning for Future Wireless Communications*. IEEE, 2020, pp. 439–457, doi: 10.1002/9781119562306.ch21.

[10] S. Zhou, W. Jadoon, and J. Shuja, "Machine learning-based offloading strategy for lightweight user mobile edge computing tasks," *Complexity*, vol. 2021, pp. 1–11, Jun. 2021.

[11] D. Gunduz, P. de Kerret, N. Sidiroupoulos, D. Gesbert, C. Murthy, and M. van der Schaar, "Machine learning in the air," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2184–2199, Oct. 2019.

[12] L. Li, G. Zhao, and R. S. Blum, "A survey of caching techniques in cellular networks: Research issues and challenges in content placement and delivery strategies," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1710–1732, 3rd Quart., 2018.

[13] J. Yao, T. Han, and N. Ansari, "On mobile edge caching," *IEEE Commun. Surveys Tuts.*, vol. 21, pp. 2525–2553, 3rd Quart., 2019.

[14] M. McClellan, C. Cervello-Pastor, and S. Sallent, "Deep learning at the mobile edge: Opportunities for 5G networks," *Appl. Sci.*, vol. 10, no. 14, p. 4735, 2020.

[15] H. Zhu, Y. Cao, W. Wang, T. Jiang, and S. Jin, "Deep reinforcement learning for mobile edge caching: Review, new features, and open issues," *IEEE Netw.*, vol. 32, no. 6, pp. 50–57, Nov. 2018.

[16] T. K. Rodrigues, K. Suto, H. Nishiyama, J. Liu, and N. Kato, "Machine learning meets computation and communication control in evolving edge and cloud: Challenges and future perspective," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 38–67, 1st Quart., 2020.

[17] S. Anokye, M. Seid, and S. Guolin, "A survey on machine learning based proactive caching," *ZTE Commun.*, vol. 17, no. 4, pp. 46–55, 2019.

[18] Y. Wang and V. Friderikos, "A survey of deep learning for data caching in edge network," *Informat.*, vol. 7, no. 4, p. 43, 2020.

[19] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 4th Quart., 2019.

[20] M. Sheraz, M. Ahmed, X. Hou, Y. Li, D. Jin, Z. Han, and T. Jiang, "Artificial intelligence for wireless caching: Schemes, performance, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 631–661, 1st Quart., 2021.

[21] J. Shuja, K. Bilal, W. Alasmary, H. Sinky, and E. Alanazi, "Applying machine learning techniques for caching in next-generation edge networks: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 181, May 2021, Art. no. 103005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804521000321

[22] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM J. Res. Develop.*, vol. 3, no. 3, pp. 210–229, 1959.

[23] N. Garg, M. Sellathurai, V. Bhatia, B. N. Bharath, and T. Ratnarajah, "Online content popularity prediction and learning in wireless edge caching," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1087–1100, Feb. 2020.

[24] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. Cambridge, MA, USA: MIT Press, 2010.

[25] H. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2017, pp. 1–10.

[26] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[27] Y. Wu, Y. Jiang, M. Bennis, F. Zheng, X. Gao, and X. You, "Content popularity prediction in fog radio access networks: A federated learning based approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[28] J. Pan and J. McElhannon, "Future edge cloud and edge computing for Internet of Things applications," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 439–449, Feb. 2018.

[29] D. Wubben, P. Rost, J. S. Bartelt, M. Lalam, V. Savin, and M. Gorgoglione, "Benefits and impact of cloud computing on 5G signal processing: Flexible centralization through cloud-RAN," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 35–44, Nov. 2014.

[30] Z. Zhou, S. Mumtaz, K. M. S. Huq, A. Al-Dulaimi, K. Chandra, and J. Rodriquez, "Cloud miracles: Heterogeneous cloud RAN for fair coexistence of LTE-U and Wi-Fi in ultra dense 5G networks," *IEEE Commun. Mag.*, vol. 56, no. 6, pp. 64–71, Jun. 2018.

[31] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[32] Z. Piao, M. Peng, Y. Liu, and M. Daneshmand, "Recent advances of edge cache in radio access networks for Internet of Things: Techniques, performances, and challenges," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1010–1028, Feb. 2019.

[33] H. Zhang, Y. Qiu, X. Chu, K. Long, and V. C. M. Leung, "Fog radio access networks: Mobility management, interference mitigation, and resource optimization," *IEEE Wireless Commun.*, vol. 24, no. 6, pp. 120–127, Dec. 2017.

[34] D. Wu, L. Zhou, Y. Cai, and Y. Qian, "Collaborative caching and matching for D2D content sharing," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 43–49, Jun. 2018.

[35] S. Gurugopinath, Y. Al-Hammadi, P. C. Sofotasios, S. Muhaidat, and O. A. Dobre, "Non-orthogonal multiple access with wireless caching for 5G-enabled vehicular networks," *IEEE Netw.*, vol. 34, no. 5, pp. 127–133, Sep. 2020.

[36] N. Zhao, F. R. Yu, L. Fan, Y. Chen, and J. Tang, "Caching unmanned aerial vehicle-enabled small-cell networks: Employing energy-efficient methods that store and retrieve popular content," *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 71–79, Mar. 2019.

[37] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[38] N. Zlatanov, A. Ikhlef, T. Islam, and R. Schober, "Buffer-aided cooperative communications: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 146–153, Apr. 2014.

[39] N. Nomikos, T. Charalambous, I. Krikidis, D. N. Skoutas, D. Vouyioukas, M. Johansson, and C. Skianis, "A survey on buffer-aided relay selection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1073–1097, 2nd Quart., 2016.

[40] L. T. Tan and R. Q. Hu, "Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10190–10203, Nov. 2018.

[41] B. Krishnamurthy and J. Rexford. (2001). *Web Protocols and Practice*. [Online]. Available: https://networking.rotocols.caching

[42] M. Reddy and G. P. Fletcher, "Intelligent web caching using document life histories: A comparison with existing cache management techniques," in *Proc. 3rd Int. Caching Workshop*, 1998, pp. 35–50.

[43] M. Abrams, C. Standridge, G. Abdulla, S. M. Williams, and E. Fox, "Caching proxies: Limitations and potentials," *World Wide Web J.*, vol. 1, pp. 1–45, Dec. 1996.

[44] M. Arlitt, R. Friedrich, and T. Jin, "Workload characterization of a web proxy in a cable modem environment," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 27, no. 2, pp. 25–36, Sep. 1999.

[45] S. Podlipnig and L. Böszörmenyi, "A survey of web cache replacement strategies," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 374–398, Dec. 2003.

[46] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[47] Z. Yang, Y. Liu, Y. Chen, and G. Tyson, "Deep reinforcement learning in cache-aided MEC networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[48] Z. Yang, Y. Liu, and Y. Chen, "Distributed reinforcement learning for NOMA-enabled mobile edge computing," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6.

[49] Z. Yang, Y. Liu, Y. Chen, and N. Al-Dhahir, "Cache-aided NOMA mobile edge computing: A reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6899–6915, Oct. 2020.

[50] A. Sadeghi, F. Sheikholeslami, A. G. Matrques, and G. B. Giannakis, "Reinforcement learning for 5G caching with dynamic cost," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 6653–6657.

[51] A. Sadeghi, F. Sheikholeslami, A. G. Marques, and G. B. Giannakis, "Reinforcement learning for adaptive caching with dynamic storage pricing," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2267–2281, Oct. 2019.

[52] A. G. Shirazi and H. Amindavar, "Channel assignment for cellular radio using extended dynamic programming," *AEU-Int. J. Electron. Commun.*, vol. 59, no. 7, pp. 401–409, Nov. 2005.

[53] C. Zhong, M. C. Gursoy, and S. Velipasalar, "Deep reinforcement learning-based edge caching in wireless networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 1, pp. 48–61, Mar. 2020.

[54] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *CoRR*, vol. abs/1509.02971, pp. 1–5, Oct. 2019.

[55] G. Dulac-Arnold, R. Evans, P. Sunehag, and B. Coppin, "Reinforcement learning in large discrete action spaces," *CoRR*, vol. abs/1512.07679, pp. 1–25, 2015.

[56] P. Wu, J. Li, L. Shi, M. Ding, K. Cai, and F. Yang, "Dynamic content update for wireless edge caching via deep reinforcement learning," *IEEE Commun. Lett.*, vol. 23, no. 10, pp. 1773–1777, Oct. 2019.

[57] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, and J. Veness, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Oct. 2015.

[58] W. Wu, Y. Gao, T. Zhou, Y. Jia, H. Zhang, T. Wei, and Y. Sun, "Deep reinforcement learning-based video quality selection and radio bearer control for mobile edge computing supported short video applications," *IEEE Access*, vol. 7, pp. 181740–181749, 2019.

[59] X. He, K. Wang, and W. Xu, "QoE-driven content-centric caching with deep reinforcement learning in edge-enabled IoT," *IEEE Comput. Intell. Mag.*, vol. 14, no. 4, pp. 12–20, Nov. 2019.

[60] M. Ma and V. W. S. Wong, "A deep reinforcement learning approach for dynamic contents caching in HetNets," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[61] M. Ma and V. W. S. Wong, "Age of information driven cache content update scheduling for dynamic contents in heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 8427–8441, Dec. 2020.

[62] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7492–7508, Nov. 2017.

[63] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1183–1210, May 2021.

[64] R. D. Yates, P. Ciblat, A. Yener, and M. Wigger, "Age-optimal constrained cache updating," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 141–145.

[65] Z. Zhang and M. Tao, "Accelerated deep reinforcement learning for wireless coded caching," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Dec. 2019, pp. 249–254.

[66] X. Xu, M. Tao, and C. Shen, "Collaborative multi-agent multi-armed bandit learning for small-cell caching," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2570–2585, Apr. 2020.

[67] X. Xu and M. Tao, "Collaborative multi-agent reinforcement learning of caching optimization in small-cell networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[68] Y. Wei, Z. Zhang, F. R. Yu, and Z. Han, "Joint user scheduling and content caching strategy for mobile edge networks using deep reinforcement learning," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[69] M. Li, F. R. Yu, P. Si, W. Wu, and Y. Zhang, "Resource optimization for delay-tolerant data in blockchain-enabled IoT with edge computing: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9399–9412, Oct. 2020.

[70] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Performance optimization for blockchain-enabled industrial Internet of Things (IIoT) systems: A deep reinforcement learning approach," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3559–3570, Jun. 2019.

[71] X. Zhang, G. Zheng, S. Lambotharan, M. R. Nakhai, and K.-K. Wong, "A reinforcement learning-based user-assisted caching strategy for dynamic content library in small cell networks," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3627–3639, Jun. 2020.

[72] A. Sengupta, S. Amuru, R. Tandon, R. M. Buehrer, and T. C. Clancy, "Learning distributed caching strategies in small cell networks," in *Proc. 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2014, pp. 917–921.

[73] K. Thar, T. Z. Oo, Y. K. Tun, D. H. Kim, K. T. Kim, and C. S. Hong, "A deep learning model generation framework for virtualized multi-access edge cache management," *IEEE Access*, vol. 7, pp. 62734–62749, 2019.

[74] K. Thar, T. Z. Oo, Z. Han, and C. S. Hong, "Meta-Learning-Based deep learning model deployment scheme for edge caching," in *Proc. 15th Int. Conf. Netw. Service Manage. (CNSM)*, Oct. 2019, pp. 1–6.

[75] *Tensorflow*. Accessed: Dec. 2020. [Online]. Available: https://www.tensorflow.org

[76] *Keras*. Accessed: Dec. 2020. [Online]. Available: https://keras.io

[77] D. Liu and C. Yang, "A deep reinforcement learning approach to proactive content pushing and recommendation for mobile users," *IEEE Access*, vol. 7, pp. 83120–83136, 2019.

[78] Z. Wang, N. D. Freitas, and M. Lanctot, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, vol. 48, Jun. 2016, pp. 1995–2003.

[79] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, vol. 48, M. F. Balcan and K. Q. Weinberger, Eds., New York, NY, USA, Jun. 2016, pp. 1928–1937.

[80] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, pp. 1–12, Jul. 2017.

[81] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Netw.*, vol. 33, no. 5, pp. 156–165, Sep. 2019.

[82] K. Guo and C. Yang, "Temporal-spatial recommendation for caching at base stations via deep reinforcement learning," *IEEE Access*, vol. 7, pp. 58519–58532, 2019.

[83] N. Garg, M. Sellathurai, V. Bhatia, and T. Ratnarajah, "Function approximation based reinforcement learning for edge caching in massive MIMO networks," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2304–2316, Apr. 2021.

[84] Y. Fang, J. Xiong, P. Cheng, and W. Zhang, "Distributed caching popular services by using deep Q-learning in converged networks," in *Proc. IEEE 90th Veh. Technol. Conf. (VTC2019-Fall)*, Jun. 2019, pp. 1–5.

[85] W. Zhang, J. Xiong, L. Gui, B. Liu, M. Qiu, and Z. Shi, "On popular services pushing and distributed caching in converged overlay networks," in *Proc. Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, 2018, pp. 1–6.

[86] P. Cheng, C. Ma, M. Ding, Y. Hu, Z. Lin, Y. Li, and B. Vucetic, "Localized small cell caching: A machine learning approach based on rating data," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1663–1676, Feb. 2019.

[87] N. Garg, M. Sellathurai, and T. Ratnarajah, "Content placement learning for success probability maximization in wireless edge caching networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3092–3096.

[88] Y. Qian, R. Wang, J. Wu, B. Tan, and H. Ren, "Reinforcement learning-based optimal computing and caching in mobile edge network," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2343–2355, Oct. 2020.

[89] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 31–37, Dec. 2017.

[90] P.-Y. Chou, W.-Y. Chen, C.-Y. Wang, R.-H. Hwang, and W.-T. Chen, "Deep reinforcement learning for MEC streaming with joint user association and resource management," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–7.

[91] W.-Y. Chen, P.-Y. Chou, C.-Y. Wang, R.-H. Hwang, and W.-T. Chen, "Live video streaming with joint user association and caching placement in mobile edge computing," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2020, pp. 796–801.

[92] B. Dai and W. Yu, "Joint user association and content placement for cache-enabled wireless access networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 3521–3525.

[93] J. Luo, F. R. Yu, Q. Chen, and L. Tang, "Adaptive video streaming with edge caching and video transcoding over software-defined mobile networks: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1577–1592, Mar. 2020.

[94] F. Wang, F. Wang, J. Liu, R. Shea, and L. Sun, "Intelligent video caching at network edge: A multi-agent deep reinforcement learning approach," in *Proc. IEEE Conf. Comput. Commun.*, Oct. 2020, pp. 2499–2508.

[95] W. Jiang, G. Feng, S. Qin, T. S. P. Yum, and G. Cao, "Multi-agent reinforcement learning for efficient content caching in mobile D2D networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1610–1622, Mar. 2019.

[96] B. Guo, X. Zhang, Q. Sheng, and H. Yang, "Dueling deep-Q-network based delay-aware cache update policy for mobile users in fog radio access networks," *IEEE Access*, vol. 8, pp. 7131–7141, 2020.

[97] G. M. S. Rahman, M. Peng, S. Yan, and T. Dang, "Learning based joint cache and power allocation in fog radio access networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4401–4411, Apr. 2020.

[98] J. Moon, O. Simeone, S.-H. Park, and I. Lee, "Online reinforcement learning of X-haul content delivery mode in fog radio access networks," *IEEE Signal Process. Lett.*, vol. 26, no. 10, pp. 1451–1455, Oct. 2019.

[99] Y. Wei, F. R. Yu, M. Song, and Z. Han, "Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor-critic deep reinforcement learning," *IEEE Internet Things J.* vol. 6, no. 2, pp. 2061–2073, Apr. 2019.

[100] J. Peters and S. Schaal, "Natural actor-critic," *Neurocomputing*, vol. 71, nos. 7–9, pp. 1180–1190, Mar. 2008.

[101] F. Jiang, J. Wang, and C. Sun, "Deep Q-learning-based cooperative caching strategy for fog radio access networks," in *Proc. Int. Conf. Commun. China (ICCC)*, 2021, pp. 922–927.

[102] L. Lu, Y. Jiang, M. Bennis, Z. Ding, F.-C. Zheng, and X. You, "Distributed edge caching via reinforcement learning in fog radio access networks," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–6.

[103] J. Yan, Y. Jiang, F. Zheng, F. R. Yu, X. Gao, and X. You, "Distributed edge caching with content recommendation in fog-rans via deep reinforcement learning," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Oct. 2020, pp. 1–6.

[104] Y. Zhou, M. Peng, S. Yan, and Y. Sun, "Deep reinforcement learning based coded caching scheme in fog radio access networks," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC Workshops)*, Aug. 2018, pp. 309–313.

[105] Y. Zhou, S. Yan, and M. Peng, "Content placement with unknown popularity in fog radio access networks," in *Proc. IEEE Int. Conf. Ind. Internet (ICII)*, Nov. 2019, pp. 361–367.

[106] Y. Sun and M. Peng, "Joint cache and radio resource management in fog radio access networks: A hierarchical two-timescale optimization perspective," in *Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2019, pp. 1–6.

[107] S. Yan, M. Jiao, Y. Zhou, M. Peng, and M. Daneshmand, "Machine-learning approach for user association and content placement in fog radio access networks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9413–9425, Oct. 2020.

[108] A. Bletsas, H. Shin, and M. Z. Win, "Cooperative communications with outage-optimal opportunistic relaying," *IEEE Trans. Wireless Commun.*, vol. 6, no. 9, pp. 3450–3460, Sep. 2007.

[109] D. Michalopoulos and G. Karagiannidis, "Performance analysis of single relay selection in Rayleigh fading," *IEEE Trans. Wireless Commun.*, vol. 7, no. 10, pp. 3718–3724, Oct. 2008.

[110] N. Nomikos, D. Poulimeneas, T. Charalambous, I. Krikidis, D. Vouyioukas, and M. Johansson, "Delay and diversity-aware buffer-aided relay selection policies in cooperative networks," *IEEE Access*, vol. 6, pp. 73531–73547, 2018.

[111] N. Nomikos, T. Charalambous, D. Vouyioukas, and G. K. Karagiannidis, "When buffer-aided relaying meets full duplex and NOMA," *IEEE Wireless Commun.*, vol. 28, no. 1, pp. 68–73, Feb. 2021.

[112] M. Radenkovic and V. S. H. Huynh, "Cognitive caching at the edges for mobile social community networks: A multi-agent deep reinforcement learning approach," *IEEE Access*, vol. 8, pp. 179561–179574, 2020.

[113] C. Zhong, M. C. Gursoy, and S. Velipasalar, "A deep reinforcement learning-based framework for content caching," in *Proc. 52nd Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2018, pp. 1–6.

[114] A. Sadeghi, G. Wang, and G. B. Giannakis, "Deep reinforcement learning for adaptive caching in hierarchical content delivery networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1024–1033, Aug. 2019.

[115] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proc. Workshop Inf.-Centric Netw.*, 2012, pp. 55–60.

[116] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Collaborative edge computing and caching with deep reinforcement learning decision agents," *IEEE Access*, vol. 8, pp. 120604–120612, 2020.

[117] P. Lin, Q. Song, J. Song, A. Jamalipour, and F. R. Yu, "Cooperative caching and transmission in CoMP-integrated cellular networks using reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5508–5520, May 2020.

[118] Z. Chen, J. Lee, T. Q. S. Quek, and M. Kountouris, "Cooperative caching and transmission design in cluster-centric small cell networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3401–3415, May 2017.

[119] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[120] X. Wang, C. Wang, X. Li, V. C. M. Leung, and T. Taleb, "Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9441–9455, Oct. 2020.

[121] W. Jiang, G. Feng, S. Qin, and Y.-C. Liang, "Learning-based cooperative content caching policy for mobile edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[122] W. Jiang, G. Feng, S. Qin, and Y. Liu, "Multi-agent reinforcement learning based cooperative content caching for mobile edge networks," *IEEE Access*, vol. 7, pp. 61856–61867, 2019.

[123] W. Jiang, G. Feng, and S. Qin, "Optimal cooperative content caching and delivery policy for heterogeneous cellular networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1382–1393, May 2017.

[124] D. Li, Y. Han, C. Wang, G. Shi, X. Wang, X. Li, and V. C. M. Leung, "Deep reinforcement learning for cooperative edge caching in future mobile networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6.

[125] W.-C. Chien, H.-Y. Weng, and C.-F. Lai, "Q-learning based collaborative cache allocation in mobile edge computing," *Future Gener. Comput. Syst.*, vol. 102, pp. 603–610, Dec. 2020.

[126] J. Sung, K. Kim, J. Kim, and J. K. Rhee, "Efficient content replacement in wireless content delivery network with cooperative caching," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Oct. 2016, pp. 547–552.

[127] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini, "Temporal locality in today's content caching: Why it matters and how to model it," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 5, pp. 5–12, Oct. 2013.

[128] J. Gu, W. Wang, A. Huang, H. Shan, and Z. Zhang, "Distributed cache replacement for caching-enable base stations in cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Mar. 2014, pp. 2648–2653.

[129] S. Chen, Z. Yao, X. Jiang, J. Yang, and L. Hanzo, "Multi-agent deep reinforcement learning-based cooperative edge caching for ultra-dense next-generation networks," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2441–2456, Apr. 2021.

[130] X. Wu, J. Li, M. Xiao, P. C. Ching, and H. V. Poor, "Multi-agent reinforcement learning for cooperative coded caching via homotopy optimization," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5258–5272, Aug. 2021.

[131] D. D. Penda, N. Nomikos, T. Charalambous, and M. Johansson, "Minimum power scheduling under Rician fading in full-duplex relay-assisted D2D communication," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2017, pp. 1–6.

[132] Z. Chen, N. Pappas, and M. Kountouris, "Probabilistic caching in wireless D2D networks: Cache hit optimal versus throughput optimal," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 584–587, Mar. 2017.

[133] P. Lin, K. S. Khan, Q. Song, and A. Jamalipour, "Caching in heterogeneous ultradense 5G networks: A comprehensive cooperation approach," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 22–32, Jun. 2019.

[134] N. Garg, M. Sellathurai, and T. Ratnarajah, "Decentralized coded caching for interference networks," in *Proc. 54th Asilomar Conf. Signals, Syst., Comput.*, Nov. 2020, pp. 1046–1050.

[135] J. Tang, H. Tang, X. Zhang, K. Cumanan, G. Chen, K.-K. Wong, and J. A. Chambers, "Energy minimization in D2D-assisted cache-enabled Internet of Things: A deep reinforcement learning approach," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5412–5423, Aug. 2020.

[136] J. Yin, L. Li, Y. Xu, W. Liang, H. Zhang, and Z. Han, "Joint content popularity prediction and content delivery policy for cache-enabled D2D networks: A deep reinforcement learning approach," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, 2018, pp. 609–613.

[137] W. Jiang, G. Feng, S. Qin, and T. S. P. Yum, "Efficient D2D content caching using multi-agent reinforcement learning," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr. 2018, pp. 511–516.

[138] T. Zhang, X. Fang, Z. Wang, Y. Liu, and A. Nallanathan, "Stochastic game based cooperative alternating Q-learning caching in dynamic D2D networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13255–13269, Dec. 2021.

[139] R. Zhang, F. R. Yu, J. Liu, R. Xie, and T. Huang, "Blockchain-incentivized D2D and mobile edge caching: A deep reinforcement learning approach," *IEEE Netw.*, vol. 34, no. 4, pp. 150–157, Oct. 2020.

[140] R. Zhang, F. R. Yu, J. Liu, T. Huang, and Y. Liu, "Deep reinforcement learning (DRL)-based device-to-device (D2D) caching with blockchain and mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6469–6485, Oct. 2020.

[141] Y. He, F. R. Yu, N. Zhao, and H. Yin, "Secure social networks in 5G systems with mobile edge computing, caching, and device-to-device communications," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 103–109, Jun. 2018.

[142] Y. He, C. Liang, F. R. Yu, and Z. Han, "Trust-based social networks with computing, caching and communications: A deep reinforcement learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 66–79, Jan. 2020.

[143] Y. He, C. Liang, F. R. Yu, and V. C. M. Leung, "Integrated computing, caching, and communication for trust-based social networks: A big data DRL approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[144] T. M. Chen and V. Venkataramanan, "Dempster–Shafer theory for intrusion detection in ad hoc networks," *IEEE Internet Comput.*, vol. 9f, no. 6, pp. 35–41, Nov. 2005.

[145] S. Bu, F. R. Yu, X. L. Peter, H. Tang, and P. Mason, "Distributed combined authentication and intrusion detection with data fusion in high security mobile ad-hoc networks," in *Proc. Mil. Commun. Conf.*, Oct. 2010, pp. 1080–1085.

[146] Z. Su and Q. Xu, "Content distribution over content centric mobile social networks in 5G," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 66–72, Jun. 2015.

[147] D. Wang, H. Qin, B. Song, X. Du, and M. Guizani, "Resource allocation in information-centric wireless networking with D2D-enabled MEC: A deep reinforcement learning approach," *IEEE Access*, vol. 7, pp. 114935–114944, 2019.

[148] S. Sun, M. Liu, Z. Jiao, X. Pang, and S. Chen, "User-centric content sharing via cache-enabled device-to-device communication," *J. Netw. Comput. Appl.*, vol. 115, pp. 103–115, Aug. 2018.

[149] M. Z. Shafiq, A. X. Liu, and A. R. Khakpour, "Revisiting caching in content delivery networks," in *Proc. Int. Conf. Meas. Model. Comput. Syst.*, 2014, vol. 42, no. 1, pp. 567–568.

[150] J. Jiang, S. Zhang, B. Li, and B. Li, "Maximized cellular traffic offloading via device-to-device content sharing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 82–91, Mar. 2016.

[151] Y. Dai, D. Xu, S. Maharjan, G. Qiao, and Y. Zhang, "Artificial intelligence empowered edge computing and caching for Internet of Vehicles," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 12–18, Jun. 2019.

[152] Y. Dai, D. Xu, K. Zhang, S. Maharjan, and Y. Zhang, "Permissioned blockchain and deep reinforcement learning for content caching in vehicular edge computing and networks," in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2019, pp. 1–6.

[153] Y. Dai, D. Xu, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4312–4324, Oct. 2020.

[154] Y. Dai, D. Xu, Y. Lu, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for edge caching and content delivery in Internet of Vehicles," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2019, pp. 134–139.

[155] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 247–257, Jan. 2020.

[156] Q. Luo, C. Li, T. H. Luan, and W. Shi, "Collaborative data scheduling for vehicular edge computing via deep reinforcement learning," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9637–9650, Oct. 2020.

[157] T. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 44–55, Jan. 2018.

[158] M. Chen, T. Wang, K. Ota, M. Dong, M. Zhao, and A. Liu, "Intelligent resource allocation management for vehicles network: An A3C learning approach," *Comput. Commun.*, vol. 151, pp. 485–494, Oct. 2020.

[159] Z. Ning, K. Zhang, X. Wang, M. S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, B. Sadoun, and R. Y. K. Kwok, ''Joint computing and caching in 5G-envisioned Internet of Vehicles: A deep reinforcement learning-based traffic control system,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 5201–5212, Aug. 2020.

[160] L. T. Tan, R. Q. Hu, and L. Hanzo, ''Twin-timescale artificial intelligence aided mobility-aware edge caching and computing in vehicular networks,'' *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3086–3099, Apr. 2019.

[161] Y. Xing, Y. Sun, L. Qiao, Z. Wang, P. Si, and Y. Zhang, ''Deep reinforcement learning for cooperative edge caching in vehicular networks,'' in *Proc. 13th Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jun. 2021, pp. 144–149.

[162] R. Wang, Z. Kan, Y. Cui, D. Wu, and Y. Zhen, ''Cooperative caching strategy with content request prediction in Internet of Vehicles,'' *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8964–8975, Jun. 2021.

[163] L. Zhao, Y. Ran, H. Wang, J. Wang, and J. Luo, ''Towards cooperative caching for vehicular networks with multi-level federated reinforcement learning,'' in *Proc. IEEE Int. Conf. Commun.*, Dec. 2021, pp. 1–6.

[164] H. Dai, H. Zhang, B. Wang, and L. Yang, ''The multi-objective deployment optimization of uav-mounted cache-enabled base stations,'' *Phys. Commun.*, vol. 34, pp. 114–120, Apr. 2019.

[165] T. Zhang, Z. Wang, Y. Liu, W. Xu, and A. Nallanathan, ''Caching placement and resource allocation for cache-enabling UAV NOMA networks,'' *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12897–12911, Nov. 2020.

[166] M. Chen, W. Saad, and C. Yin, ''Echo-liquid state deep learning for 360° content transmission and caching in wireless VR networks with cellular-connected UAVs,'' *IEEE Trans. Commun.*, vol. 67, no. 9, pp. 6386–6400, Mar. 2019.

[167] M. Bennis and D. Niyato, ''A Q-learning based approach to interference avoidance in self-organized femtocell networks,'' in *Proc. IEEE Globecom Workshops*, Dec. 2010, pp. 706–710.

[168] M. Chen, W. Saad, and C. Yin, ''Virtual reality over wireless networks: Quality-of-service model and learning-based resource management,'' *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5621–5635, Nov. 2018.

[169] M. Chen, W. Saad, and C. Yin, ''Liquid state machine learning for resource allocation in a network of cache-enabled LTE-U UAVs,'' in *Proc. IEEE Global Commun. Conf.*, Oct. 2017, pp. 1–6.

[170] M. Chen, W. Saad, and C. Yin, ''Liquid state machine learning for resource and cache management in LTE-U unmanned aerial vehicle (UAV) networks,'' *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1504–1517, Mar. 2019.

[171] A. Al-Hilo, M. Samir, C. Assi, S. Sharafeddine, and D. Ebrahimi, ''UAV-assisted content delivery in intelligent transportation systems-joint trajectory planning and cache management,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5155–5167, Aug. 2020.

[172] C. Wu, S. Shi, S. Gu, L. Zhang, and X. Gu, ''Deep reinforcement learning-based content placement and trajectory design in urban cache-enabled uav networks,'' *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–11, Oct. 2020.

[173] N. Nomikos, T. Charalambous, D. Vouyioukas, R. Wichman, and G. K. Karagiannidis, ''Power adaptation in buffer-aided full-duplex relay networks with statistical CSI,'' *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7846–7850, Aug. 2018.

[174] T. Charalambous, S. M. Kim, N. Nomikos, M. Bengtsson, and M. Johansson, ''Relay-pair selection in buffer-aided successive opportunistic relaying using a multi-antenna source,'' *Ad Hoc Netw.*, vol. 84, pp. 29–41, Mar. 2019.

[175] R. Simoni, V. Jamali, N. Zlatanov, R. Schober, L. Pierucci, and R. Fantacci, ''Buffer-aided diamond relay network with block fading and inter-relay interference,'' *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7357–7372, Nov. 2016.

[176] H. Zhang, Y. Li, D. Jin, M. M. Hassan, A. Alelaiwi, and S. Chen, ''Buffer-aided device-to-device communication: Opportunities and challenges,'' *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 67–74, Dec. 2015.

[177] S. Maghsudi and E. Hossain, ''Multi-armed bandits with application to 5G small cells,'' *IEEE Wireless Commun.*, vol. 23, no. 3, pp. 64–73, Jun. 2016.

[178] N. Nomikos, S. Talebi, R. Wichman, and T. Charalambous, ''Bandit-based relay selection in cooperative networks over unknown stationary channels,'' in *Proc. IEEE 30th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Sep. 2020, pp. 1–6.

[179] S. Ghoorchian and S. Maghsudi, ''Multi-armed bandit for energy-efficient and delay-sensitive edge computing in dynamic networks with uncertainty,'' *IEEE Trans. Cognit. Commun. Netw.*, vol. 7, no. 1, pp. 279–293, Mar. 2021.

[180] N. Nomikos, T. Charalambous, and R. Wichman, ''Bandit-based power control in full-duplex cooperative relay networks,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–6.

[181] A. Dimas, K. Diamantaras, and A. P. Petropulu, ''Q-learning based predictive relay selection for optimal relay beamforming,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 5030–5034.

[182] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, ''A deep learning framework for optimization of MISO downlink beamforming,'' *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1866–1880, Mar. 2020.

[183] C. Psomas and I. Krikidis, ''Wireless powered mobile edge computing: Offloading or local computation?'' *IEEE Commun. Lett.*, vol. 24, no. 11, pp. 2642–2646, Nov. 2020.

[184] Z. Ding, P. Fan, G. K. Karagiannidis, R. Schober, and H. V. Poor, ''NOMA assisted wireless caching: Strategies and performance analysis,'' *IEEE Trans. Commun.*, vol. 66, no. 10, pp. 4854–4876, Oct. 2018.

[185] L. Xiang, D. W. K. Ng, X. Ge, Z. Ding, V. W. S. Wong, and R. Schober, ''Cache-aided non-orthogonal multiple access: The two-user case,'' *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 3, pp. 436–451, Jun. 2019.

[186] X. Pei, H. Yu, Y. Chen, M. Wen, and G. Chen, ''Hybrid multicast/unicast design in NOMA-based vehicular caching system,'' *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16304–16308, Dec. 2020.

[187] S. Luo and K. C. Teh, ''Adaptive transmission for cooperative NOMA system with buffer-aided relaying,'' *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 937–940, Apr. 2017.

[188] N. Nomikos, T. Charalambous, D. Vouyioukas, R. Wichman, and G. K. Karagiannidis, ''Integrating broadcasting and NOMA in full-duplex buffer-aided opportunistic relay networks,'' *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9157–9162, Aug. 2020.

[189] P. Xu, Y. Wang, G. Chen, G. Pan, and Z. Ding, ''Design and evaluation of buffer-aided cooperative NOMA with direct transmission in IoT,'' *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8145–8158, May 2021.

[190] J. Li, X. Lei, P. D. Diamantoulakis, F. Zhou, P. Sarigiannidis, and G. K. Karagiannidis, ''Resource allocation in buffer-aided cooperative non-orthogonal multiple access systems,'' *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7429–7445, Dec. 2020.

[191] E. C. Strinati and S. Barbarossa, ''6G networks: Beyond Shannon towards semantic and goal-oriented communications,'' *Comput. Netw.*, vol. 190, 2021, Art. no. 107930.

[192] M. Kountouris and N. Pappas, ''Semantics-empowered communication for networked intelligent systems,'' *IEEE Commun. Mag.*, vol. 59, no. 6, pp. 96–102, Jun. 2021.

[193] Q. Lan, D. Wen, Z. Zhang, Q. Zeng, X. Chen, P. Popovski, and K. Huang, ''What is semantic communication? A view on conveying meaning in the era of machine intelligence,'' 2021, *arXiv:2110.00196*.

[194] W. Zhao, Z. Chen, K. Li, N. Liu, B. Xia, and L. Luo, ''Caching-aided physical layer security in wireless cache-enabled heterogeneous networks,'' *IEEE Access*, vol. 6, pp. 68920–68931, 2018.

[195] T. X. Zheng, H.-M. Wang, and J. Yuan, ''Secure and energy-efficient transmissions in cache-enabled heterogeneous cellular networks: Performance analysis and optimization,'' *IEEE Trans. Wireless Commun.*, vol. 66, no. 11, pp. 5554–5567, Nov. 2018.

[196] L. Dong, Z. Han, A. P. Petropulu, and H. V. Poor, ''Improving wireless physical layer security via cooperating relays,'' *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1875–1888, Mar. 2010.

[197] A. Garnaev, A. Petropulu, W. Trappe, and H. V. Poor, ''A multi-jammer game with latency as the User's communication utility,'' *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 1899–1903, Sep. 2020.

[198] Z. Yu, J. Hu, G. Min, Z. Wang, W. Miao, and S. Li, ''Privacy-preserving federated deep learning for cooperative hierarchical caching in fog computing,'' *IEEE Internet Things J.*, early access, May 18, 2021, doi: 10.1109/JIOT.2021.3081480.

[199] G. Shi, Y. Xiao, Y. Li, and X. Xie, ''From semantic communication to semantic-aware networking: Model, architecture, and open problems,'' *IEEE Commun. Mag.*, vol. 59, no. 8, pp. 44–50, Aug. 2021.

[200] N. Garg, M. Sellathurai, and T. Ratnarajah, ''In-network caching for hybrid satellite-terrestrial networks using deep reinforcement learning,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 8797–8801.

NIKOLAOS NOMIKOS (Senior Member, IEEE) received the Diploma degree in electrical engineering and computer technology from the University of Patras, Greece, in 2009, and the M.Sc. and Ph.D. degrees from the Information and Communication Systems Engineering Department, University of the Aegean, Samos, Greece, in 2011 and 2014, respectively. From November 2014 to October 2019, he worked as a Postdoctoral Researcher with the Information and Communication Systems Engineering Department, University of the Aegean. From September 2018 to September 2019, he was an Adjunct Lecturer at the Open University of Cyprus. From January 2019 to December 2019, he worked as a Postdoctoral Researcher with the General Department, National and Kapodistrian University of Athens. He is currently a Research Associate with the IRIDA Research Centre for Communication Technologies, Department of Electrical and Computer Engineering, University of Cyprus. His research interests include cooperative communications, non-orthogonal multiple access, full-duplex communications, and machine learning for wireless networks optimization. He is a member of the IEEE Communications Society and the Technical Chamber of Greece.

SPYROS ZOUPANOS received the B.Sc. and M.Sc. degrees from the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, in 2004 and 2006, respectively, and the Ph.D. degree from the INRIA Saclay, Université Paris-Sud (Université Paris-Saclay), in 2009. In the past, he was a Collaborating Researcher with the Athena Research Center (ARC), a Researcher and a Software Engineer at École Polytechnique Fédérale de Lausanne (EPFL), a Senior Developer at Quartet Financial Systems (now ActiveViam) and a Postdoctoral Fellow at Max-Planck-Institut für Informatik. Currently, he is a Research Associate with the Department of Informatics, Ionian University, and a Software Engineer at the Independent Authority for Public Revenue (IAPR). His research interests include distributed data management and big data, databases, analytics, mobile networks, knowledge extraction, and machine learning.

THEMISTOKLIS CHARALAMBOUS (Senior Member, IEEE) received the B.A. degree in electrical and information sciences from Trinity College Dublin, the M.Eng. degree in electrical and information sciences from the University of Cambridge, in 2005, and the Ph.D. degree from the Control Laboratory, Engineering Department, University of Cambridge, in 2010. He joined the Human Robotics Group, Imperial College London, as a Research Associate, from September 2009 to September 2010. From September 2010 to December 2011, he worked as a Visiting Lecturer with the Department of Electrical and Computer Engineering, University of Cyprus. From January 2012 to January 2015, he worked with the Department of Automatic Control, School of Electrical Engineering, Royal Institute of Technology, as a Postdoctoral Researcher. From April 2015 to December 2016, he worked as a Postdoctoral Researcher with the Department of Electrical Engineering, Chalmers University of Technology. In January 2017, he joined the Department of Electrical Engineering and Automation, School of Electrical Engineering, Aalto University, as a tenure-track Assistant Professor. Since September 2018, he has been a Research Fellow of the Academy of Finland. From July 2020 to August 2021, he was a tenured Associate Professor and since then he has been a Visiting Professor. Since September 2021, he has been a tenure-track Assistant Professor at the University of Cyprus. His primary research interests include the design and analysis of (wireless) networked control systems that are stable, scalable, and energy efficient.

IOANNIS KRIKIDIS (Fellow, IEEE) received the Diploma degree in computer engineering from the Computer Engineering and Informatics Department (CEID), University of Patras, Greece, in 2000, and the M.Sc. and Ph.D. degrees from École Nationale Supérieure des Télécommunications (ENST), Paris, France, in 2001 and 2005, respectively, all in electrical engineering.

From 2006 to 2007, he worked as a Postdoctoral Researcher with ENST, and from 2007 to 2010, he was a Research Fellow with the School of Engineering and Electronics, The University of Edinburgh, Edinburgh, U.K. He is currently an Associate Professor at the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus. His current research interests include wireless communications, cooperative networks, 5G/B5G communication systems, wireless powered communications, and intelligent reflected surfaces. He was the recipient of the 2013 Young Researcher Award from the Research Promotion Foundation, Cyprus, the 2016 IEEEComSoc Best Young Professional Award in Academia, and the 2019 IEEE Signal Processing Letters Best Paper Award. He has been recognized by the Web of Science as a Highly Cited Researcher, from 2017 to 2021. He has received the prestigious ERC Consolidator Grant. He serves as an Associate Editor for IEEE Transactions on Wireless Communications, IEEE Transactions on Green Communications and Networking, and IEEE Wireless Communications Letters.