Artificial Intelligence for Wireless Caching: Schemes, Performance, and Challenges

Muhammad Sheraz, Manzoor Ahmed, Xueshi Hou, Yong Li, *Senior Member, IEEE*, Depeng Jin, *Member, IEEE*, Zhu Han, *Fellow, IEEE*, and Tao Jiang, *Fellow, IEEE*

***Abstract*—Wireless data traffic is growing unprecedentedly and it may impede network performance by consuming an ever-greater amount of bandwidth. With the advancement in technology there exist profound techniques having potentials to improve performance of wireless networks. Artificial Intelligence (AI) is one such evolving technology that enables systems to take intelligent decisions. AI can be incorporated in wireless networks for performing an optimal data caching based on accurate predictions of users' data requests and data popularity profile. AI-based data caching is a promising candidate to effectively harness the issues of rising backhaul data traffic of future wireless networks such as duplicate data transmission and data access delay. In this paper, we provide a systematic survey of state-of-the-art intelligent data caching approaches based on learning mechanism to optimize data caching. First we give an overview of traditional caching approaches and their limitations. Then, after rendering brief introduction of several AI techniques, we introduce state-of-the-art learning approaches in cache-enabled wireless networks. We unfold significant research efforts utilizing AI for efficient data placement for optimizing network performance in terms of cache hit rate, throughput, and offloading etc. Finally, we highlight existing challenges and research directions of AI-based data caching.**

***Index Terms*—Caching, artificial intelligence, intelligent data caching, supervised learning, unsupervised learning, reinforcement learning, transfer learning.**

I. Introduction

WIRELESS networks are predicted to grow tremendously in the years to come and it is expected that by 2020 there will be a 200 percent rise in the mobile data traffic [1]. According to the report, the number of mobile devices is expected to touch 11.5 billion and data traffic will rise approximately to 367 EB by the year 2020 [2], which is mainly driven by increasing demand from social networking and video streaming. The Fifth Generation (5G) cellular network is envisioned to efficiently handle growing data demands and Key Performance Indicators (KPIs) including latency, mobility, and capacity [3]. This requires a wide range of innovative techniques to cope with the future challenges of wireless networks.

Since caching has already shown tremendous benefits in wired networks to reduce latency and conserve energy, it is appears promising to implement the data caching in wireless networks. Among various solutions, one approach called Stochastic Collaborative Content Placement (SCCP) caching technique is to bring data close to the end users, which may avoid unnecessary delay and network overhead. Specifically, this technique can be exploited to optimize data caching with Time-to-Live (TTL) scheme, which ensures cache occupancy time of each content while considering stochastic behavior of new arriving contents [4]. Moreover, the data can be cached in User Equipments' (UEs) and/or Base Statinos' (BSs) cache. Moving data close to users, by exploiting infrastructure caching (i.e., utilizing BSs) or infrastructureless caching (i.e., utilizing UEs), seems to be an efficient approach for reducing delay and alleviating network overhead [5], [6]. In an infrastructure based caching, deployment of cache entities such as BSs or helper nodes are required to avoid content fetching from the core network. However, this may incur substantial Capital Expenditure (CAPEX). Conversely, in an infrastructureless caching, contents can be cached in UEs' caches that do not require any CAPEX and also provide an opportunity to exploit Device-to-Device (D2D) communications for data sharing among users [7], [8].

Data caching at BSs can alleviate network congestion by avoiding data access from the core network. Specifically, these BSs can share data with each other via wired or wireless link

Manuscript received June 26, 2019; revised December 31, 2019 and March 27, 2020; accepted May 13, 2020. Date of publication July 10, 2020; date of current version February 24, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB180003400; in part by the National Nature Science Foundation of China under Grant U1936217, Grant 61971267, Grant 61972223, Grant 61941117, and Grant 61861136003; in part by Beijing Natural Science Foundation under Grant L182038; in part by the Beijing National Research Center for Information Science and Technology under Grant 20031887521; in part by the research fund of Tsinghua University—Tencent Joint Laboratory for Internet Innovation Technology; in part by the U.S. MURI AFOSR MURI under Grant 18RT0073; and in part by NSF under Grant EARS-1839818, Grant CNS1717454, Grant CNS-1731424, and Grant CNS-1702850. *(Corresponding author: Yong Li.)*

Muhammad Sheraz, Yong Li, and Depeng Jin are with the Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: xlc16@mails.tsinghua.edu.cn; liyong07@tsinghua.edu.cn; jindp@tsinghua.edu.cn).

Manzoor Ahmed is with the College of Computer Science and Technology, Qingdao University, Qingdao 266071, China (e-mail: manzoor.achakzai@gmail.com).

Xueshi Hou is with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: x7hou@ucsd.edu).

Zhu Han is with the Computer Science Department, University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 446-701, South Korea (e-mail: zhan2@uh.edu).

Tao Jiang is with the Wuhan National Laboratory for Optoelectronics and School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: Tao.Jiang@ieee.org).

Digital Object Identifier 10.1109/COMST.2020.3008362

TABLE I
COMPARISON OF OUR WORK WITH RELATED SURVEY ARTICLES

Existing works	Year	Journal	AI approach	Contribution
[5]	2015	IEEE Communications Surveys & Tutorials	✘	Provided overview of in-network caching techniques for ICN. Moreover, strengths and weakness of in-network caching mechanisms are elaborated. Also, simulation based comparative analysis and performance affecting factors of caching mechanisms are discussed.
[26]	2016	IEEE Communications Surveys & Tutorials	✘	Discussed on-path caching and data forwarding techniques in ICN, and provided taxonomy for on-path caching and highlighted the trends and evaluation issues.
[29]	2017	IEEE Communications Surveys & Tutorials	✘	Developed taxonomy for cache discovery, surveyed and classified the associated cache discovery techniques of MANET-based cooperative caching schemes. Analyzed the potential in addressing the specific challenges that occur when deploying non-safety applications within VANETs.
[30]	2017	IEEE Access	✘	An overview of mobile edge networks and a comprehensive survey on caching, computation, and communication techniques are presented.
[27]	2018	IEEE Communications Surveys & Tutorials	✘	Surveyed cache management techniques in ICN. Performance is evaluated in a simulation network environment.
[7]	2018	IEEE Communications Surveys & Tutorials	✘	A tutorial on socially aware D2D communications is provided. Categorization of D2D applications into data offloading, caching, and data dissemination is performed.
[28]	2019	IEEE Internet of Things Journal	✘	Provided overview of hierarchical edge caching structure in radio access networks for IoT. The key caching approaches are presented based on the deployment location of edge caches, content placement strategies, and coded caching.
[31]	2019	IEEE Communications Surveys & Tutorials	Supervised learning, unsupervised learning, reinforcement learning, deep neural networks, and transfer learning	A comprehensive survey on advancement of learning based resource management, networking, and mobility management. Research works are categorized based on the different learning techniques.
[32]	2019	IEEE Communications Surveys & Tutorials	Deep reinforcement learning	A complete DRL tutorial and its applications in communications and networking are provided. Surveyed several DRL approaches to deal with emerging challenges in data communications.
[33]	2019	IEEE Communications Surveys & Tutorials	Artificial Neural Networks (ANNs)	A complete tutorial on ANNs and its applications in wireless networks. Discussed ANNs techniques for solving problems in wireless communications.
Our article	–	–	Supervised learning, unsupervised learning, reinforcement learning, deep neural networks, and transfer learning	An overview of data caching including caching structures, policies, and challenges is given. Systematically overview the AI techniques. Followed by a comprehensive survey considering AI based data caching in wireless networks. Discussed AI coupled challenges and future research directions for optimizing wireless network performance.

(e.g., millimeter wave (mmWave)) to increase capacity tremendously [9], [10]. For instance, contents can be cached in a BS and whenever any user within its coverage area demand for a content, its requested content is served with an immediate effect. Moreover, apart from BSs, UEs are also equipped with limited cache resources. Since there is a constraint on UEs cache capacity, popular contents following global popularity or user preferences should be placed in their cache resources. It enables users to locate desired contents within their own cache or in the cache of their neighboring UEs that exist within D2D communication range. In 5G, D2D communications is considered to play a pivotal role in reducing backhaul data traffic load by leveraging direct communications among proximity users [11]. This approach has the potential to improve data rate, energy conservation, spectral efficiency, and network's capacity. However, D2D communications occurs at the cost of users' scarce resources (i.e., memory and power). As a common perception, users are selfish by nature and do not want to participate in data sharing at the cost of their resources consumption such as device battery and limited cache. Therefore, certain incentive mechanisms in terms of monetary or exclusive services are required to motivate them to play their role in improving systems' utility [12]–[14]. Most of the data traffic constitutes of popular contents alone and it keeps evolving with respect to time. Social and geographic data correlation, and users' previous data history can be used to know about data evolving patterns [15]. It is essential to accurately predict data popularity profile and perform data placement [16], [17].

There is a serious need to design efficient caching mechanisms while considering limited cache resources, dynamic data popularity profile, users' mobility patterns, and data access behavior. To improve accuracy of prediction model, it is imperative to process a large amount of information such as contents features, cache location, and cache technique including proactive and reactive caching. For instance, in proactive caching, contents are cached prior to users' content requests. Conversely, in reactive caching contents are cached following the users' data demands. This makes accurate prediction an intractable problem. In general, traditional schemes for data caching are becoming ineffective due to significant growth in the data traffic. New caching solutions are needed to handle extensive data traffic by considering user data demands, channel conditions, mobility patterns, and variations in data popularity.

AI has the potential to play significant role in several domains of wireless networks [18]–[24]. It can exploit big data analytics to improve the prediction of users' social behaviors to improve network performance. The wireless network has to analyze a large amount of data generated by a massive number of users available in the network. AI can enable intelligent resource management in contrast to traditional optimization approaches. The network states can be learned in real time by utilizing AI, which is beneficial in decision making in a dynamic environment. According to intelligence level, AI approaches can be partitioned into two categories. The first type is basic that enables system to respond the environment in a deterministic way. It enables network to configure based on the performance indicators. The second type is based on the complete capability of system to interact with the environment, which enables system to take decisions even in case of unknown environment [25]. Advancement in AI has shown tremendous improvement in different scientific domains such as robotics, medical, and space sciences, etc. AI can be a solution to deal with a huge amount of information because it is capable to conduct learning process on high-dimensional input dataset. Therefore, AI potential needs to be utilized for increasing the utility of wireless networks. In this paper, first we discuss conventional approaches for data caching. Then, we discuss AI algorithms and their usage in improving the performance of data caching mechanisms.

A. Previous Works

Considering the significant potential and recent advancement in AI and the role of AI in wireless networks, it is timely to survey AI-based data caching schemes. Although numerous survey articles on conventional data caching techniques without leveraging AI have been published in recent years, none of them had focused on the AI-based data caching in wireless communications. Regarding traditional caching techniques, Zhang *et al.* in [5] discussed the architecture of Information-Centric Networking (ICN) and provided an insight to the caching mechanism in ICN. They have given a performance comparison between cooperative and non-cooperative caching to observe cache hit ratio. Followed by Ioannou and Weber in [26], the authors focused on-path caching in ICN and conducted in-depth analytical discussion on currently employed caching schemes and forwarding approaches. While Din *et al.* [27] endeavored to provide detail discussion of cache management techniques in ICN. Different from [5], [26] and [27], Piao *et al.* in [28] surveyed the data placement strategies at network edge to optimize network performance in terms of network latency, spectral, and energy efficiency. Other than ICN, Glass *et al.* [29] provided a comprehensive survey of Mobile Ad Hoc Networks (MANETs) based cooperative caching approaches and categorized cache discovery mechanisms. Similarly, Ahmed *et al.* [7] introduced social-awareness based data caching in D2D communications. For mobile edge networks, Wang *et al.* [30] surveyed computation, caching, and communication mechanisms. Regarding AI-based surveys, Sun *et al.* [31] discussed a comprehensive survey of learning-based approaches in networking, mobility, and resource management fields. Also, Luong *et al.* [32] presented a tutorial on DRL, and put forward a detail discussion on addressing communication and networking issues using DRL. Similarly, Chen *et al.* [33] provided an analysis on different kinds of ANN and how to solve edge caching and computing problem in wireless networks. Since there is no comprehensive survey on AI-based caching in wireless networks, there is a realization of a need of one such survey article providing an insight of AI-based caching schemes in wireless networks. Therefore, we conduct our efforts to provide comprehensive and in-depth knowledge about AI technology and its benefits in wireless networks. This survey aims to help the researchers to work in the direction of AI-based data caching in cache-enabled wireless networks.

B. Contributions

We focus primarily on AI-based caching techniques using supervised learning, unsupervised learning, reinforcement learning, and transfer learning. After providing a detail discussion of their implementation and improvement in network performance, we highlight existing challenges and future research directions. The contributions of our article are the following:

- We provide a brief highlight of existing caching approaches in wireless networks. This will help new researchers in the field of data caching to grasp key concepts of data caching in wireless networks.
- We provide an overview of several AI algorithms that will facilitate readers in the field of AI, and the fundamental idea of AI technology.
- A detail discussion of several algorithms including supervised, unsupervised, Reinforcement Learning (RL), and Transfer Learning (TL) regarding their implementation and impact on cache performance is provided. To increase the readability and dwelling, we organize the survey according to the network's performance metrics including offloading, cache hit rate, delay, throughput, and cost.
- The challenges and potential research directions are identified for AI-based caching in wireless networks.

The structure of our paper is as follows. In Section II, we provide overview of traditional caching approaches including

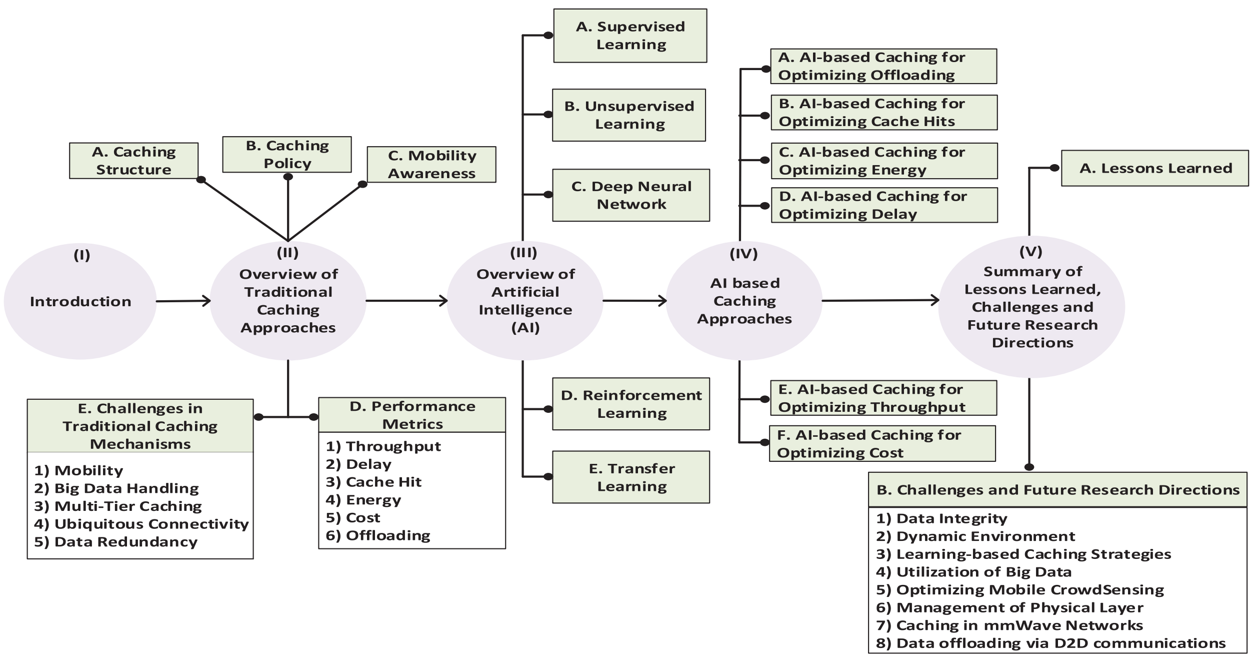

Fig. 1. Tutorial structure of AI-based cache enabled wireless networks.

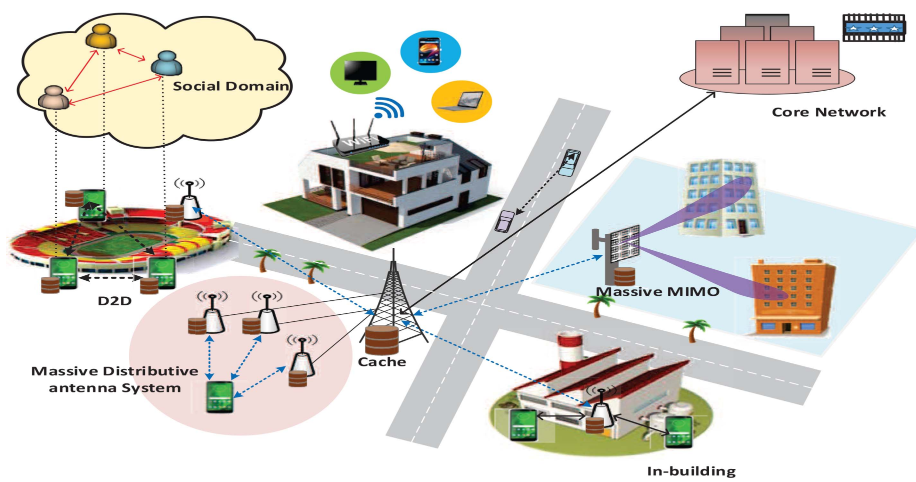

Fig. 2. An illustration of cache-enabled wireless networks.

performance metrics and its challenges. In Section III, we provide an overview of AI including supervised, unsupervised, RL, and TL techniques. In Section IV, we discuss AI-based caching approaches. In Section V, we discuss the lessons learned, existing challenges, and future research directions. Finally, Section VI concludes the paper. Moreover, Fig. 1 illustrates the structure of our survey. The definitions of acronyms are summarized in Table II for convenience of reference.

II. Overview of Traditional Caching Approaches

The fundamental concept of caching is data placement in various locations to offload data traffic through cache resources, which can reduce network congestion. In case of a wired network, users generate data requests and if the requested contents are available at a router, then users' data demands are fulfilled with low delay. However, if contents are not available at router, then data requests are forwarded to server by traversing through the core network and contents are provided to the users [34]. In case of a wireless network, users and BSs are equipped with cache capabilities. In such scenario, user's data content request is full-filled from its own cache or neighboring UEs through D2D communications. If the content is not cached by the user itself or neighboring UEs, then it can be obtained from the serving BS. Otherwise, content is fetched from the core network by the serving BS, and then provided to the content demanding user.

Data caching is a promising technique to meet diverse data demands, nevertheless, optimizing data service is a critical part for data caching. In this regard, few works [35], [36] exploited recommendation system, however, it may consume scarce energy and cache resources while performing computation at the edge. According to [37], network coding can solve energy and cache consumption shortcomings. The authors considered the two fundamental types (i.e., straight network

TABLE II
LIST OF ACRONYMS

Acronym	Definition	Acronym	Definition
ADMM	Alternating Direction Method of Multipliers	LFA	Linear Function Approximation
AI	Artificial Intelligence	LISO	Longest Lifetime In-Shortest Lifetime Out
BBUs	Base Band Units	LRM	Likelihood-Ratio Method
BMU	Best Matching Unit	LSTM	Long Short Term Memory
CA	College Admission Caching	MAB	Multi-Arm Bandit
CAPEX	Capital Expenditure	MANETs	Mobile Ad Hoc Networks
CBPMF	Constrained Bayesian Probabilistic Matrix Factorization	MCUCB	Modified Combinatorial Upper Confidence Bounds
CMAB	Combinatorial Multi-Arm Bandit	MDP	Markov Decision Process
CNN	Convolutional Neural Network	MDS	Maximum-Distance Separable
C-RAN	Cloud Radio Access Network	MENs	Mobile Edge Networks
CUCB	Combinatorial Upper Confidence Bounds	MLPs	Multi-Layer Perceptrons
CUCBSC	Combinatorial Upper Confidence Bounds with Switching Cost	MSNs	Mobile Social Networks
D2D	Device to Device communications	NN	Neural Networks
DBN	Deep Belief Network	OGD	Online Gradient Descent
DCA	Deterministic Caching Algorithm	PAC	Probably Approximately Correct
DDPG	Deep Deterministic Policy Gradient	QoE	Quality of Experience
DL-CPP	Deep-Learning Content Popularity Prediction	RBF	Radial Based Function
DGPA-SC	Discrete Generalized Pursuit Algorithm with Social Characters	RL	Reinforcement Learning
DNN	Deep Neural Network	RLMA	Reinforcement Learning based Model-Free Acceleration
DRL	Deep Reinforcement Learning	RNN	Recurrent Neural Network
EE	Energy Efficiency	RRHs	Radio Resource Heads
ESNs	Echo State Networks	RSUs	Road Side Units
FDM	Finite Difference Method	SAE	Stacked Auto-Encoders
FIFO	First-In First-Out	SBS	Small Base Stations
FSMC	Finite-State Markov Channel	SE	Spectral Efficiency
GOL	Generic Online Learning	SOMs	Self-Organizing Maps
HetNets	Heterogeneous Networks	SPMAB	Single Player Multi-Arm Bandit
ICN	Information-Centric Networking	SVM	Support Vector Machine
ICRP	Individual Content Request Probability	TL	Transfer Learning
IoT	Internet of Thing	UAV	Unmanned Aerial Vehicle
JAL MCUCB	Joint Action Learners MCUCB	UDNs	Ultra Dense Networks
k-NN	k-Nearest Neighbor	WCDNs	Wireless Content Delivery Networks
LECC	Learning based Cooperative Caching	WIN	Wireless Infostation Network

coding and physical layer network coding) of network coding. In the straight network coding, the data source $x1$ broadcasts $s1$ to BS, or relay r, and user $z1$ in the first time slot, while, in the second time slot, data source $x2$ broadcasts $s2$ to relay r and user $z2$. Next, relay r performs XOR on $s1$ and $s2$ bit by bit, and forward the data to users $z1$ and $z2$, respectively. In physical layer network coding, data sources $x1$ and $x2$ broadcast $s1$ and $s2$ to relay r simultaneously, which then mixes the received data signals, and broadcasts the mixed signal to users' $z1$ and $z2$. The advantage of physical layer network coding over straight network coding is that it requires less time slots, which greatly reduces waiting time to receive data. Network coding has been widely for information spreading and network throughput maximization [38]. There are two types of network coding, i.e., intra-session network coding [39] and inter-session network coding [40]. Intra-session network coding is utilized for bulk data transmission by dividing into several segments. However, users that receive incomplete segments devise linear network coding for successful data download. However, inter-session network coding makes X topology network that combines contents from various transmitters and transmit to various users.

Moreover, the benefits of wireless network caching can be improved by combining data caching with the physical layer communication schemes [41]–[44]. In this case, the main idea is to perform transmission link modeling while considering data caching. This enables physical layer aware data caching providing multi-user diversity gains by performing Channel State Information (CSI)-aware data transmission scheduling [45], and achieves substantial reduction in latency. One such effort is carried out in [46], where cooperative Multiple-Input Multiple-Output (MIMO) transmission scheme is combined with data placement at multiple BSs, and data services are provided to the users in a cooperative manner. Thus, physical layer aware data caching in wireless networks can provide significant performance gains while guaranteeing high energy efficiency and QoE [47], such as low data streaming latency that is dependent on the data delivery time. An illustration of cache-enabled wireless network is provided in Fig. 2, where contents are cached at the BS and Small Base Stations (SBSs) in order to avoid data fetching from the core network. Moreover, content sharing among UEs is also depicted to optimize network performance. In the following subsections, we discuss caching structure and caching policies including coded and uncoded data caching. Then, we discuss impact of mobility on the caching mechanism performance. Afterwards, we point out performance metrics for evaluation of caching mechanisms. Finally, we highlight existing challenges in the traditional caching mechanisms.

A. Caching Structure

There are two fundamental structures of caching policies. The data placement can be implemented either in a centralized or distributed manner, which are summarized as:

Centralized Caching: In centralized caching, a single entity (e.g., BS) gathers all the underlying network information regarding content popularity, CSI, and data demands. The data content is placed in the cache resources after acquiring all the aforementioned necessary information [48]. Thus, centralized caching complexity is very high following diverse network characteristics.

Distributed Caching: In distributed caching, cache entities makes cache decision in an independent manner regarding contents to be cached and does not require whole network information that infers reduction in complexity [49]. Distributed caching has lower complexity than centralized caching. However, distributed caching based algorithms may not achieve global optimal solution because all decisions are taken independently by the cache entities. Therefore, it is hard to anticipate the cache status of other entities that leads to degradation in cache performance.

B. Caching Policy

There are two types of data caching approaches. One is coded data caching and the other is uncoded data caching. The difference between them is whether to perform data segmentation by using the data coding (e.g., fountain coding or network coding) or not.

Coded Data Caching: In coded data caching, the data is operated via algebraic algorithms. After performing data encoding, contents are placed in the cache resources. When a user demands a content, encoded data is sent through the cache resources that is decoded at the user end to retrieve original data. Coded data coding optimizes flow of data in a wireless network by accumulating multiple transmissions. This helps to eliminate duplicate data transmissions, which improves energy conservation and spectral efficiency [50]–[52].

Uncoded Data Caching: Uncoded data caching stores complete contents in the cache resources [47], [53], [54]. This makes a feasible solution to intermittent network connectivity, where all users are mobile and objective is to transfer contents within a specific period of time.

Both coded and uncoded caching policies are helpful in improving the network utility in terms of increasing cache hit rate, throughput, and reducing cost. Moreover, the cached contents are required to be replaced with new data having high popularity because such contents are generating source of huge data traffic. Several techniques have been designed for cache replacement problem like recency-based caching and frequency-based caching that replaces Least Recently Used (LRU) and Least Frequently Used (LFU) contents with new contents, respectively [55]–[57]. The evolving popularity of contents is taken into account in order to perform optimal data caching because users' data requests are skewed towards popular contents [58], [59].

C. Mobility Awareness

The most important factor having significant impact on network performance is users' mobility. In a dynamic network scenario, for appropriate data placement it is essential to consider users' mobility patterns. A dynamic network topology leads to services' uncertainties due to which caching becomes a challenging task [60]. Users' positions can be utilized to achieve optimal caching. However, information regarding trajectory is unknown and it requires a robust content placement policy. The user mobility can be categorized into spatial and temporal properties. The former is related to physical mobility patterns of users and the latter is relevant to the communication time duration. The modeling of user locations can be accomplished in three ways. First is the Poisson Point Process (PPP)-based models [54], [61]–[65], where homogeneous PPP modeling is used for users locations. Second is the Poison based clustering models [66] that groups together all the users in various clusters. Each cluster contains a cluster center known as centroid following PPP, where the position of cluster members is modeled by normal distribution. Third is the grid-based clustering models [67]–[70], which assume users' locations are at the grids intersections. After determining mobility pattern of each user at any given time, stringent content placement mechanisms can be used for optimizing the network performance. However, users behave selfishly and are unwilling to improve network performance at the cost of consuming their own resources such as energy and storage capacity. Thus, incentive schemes and social relationships including social tie and common interests can be used to motivate users to play their role in improving network utility [15], [51], [69]–[72]. Ultimately, the network operators can benefit from increasing users by gathering their cache resources for storing the number of contents. This may cause less network overhead within the existing network infrastructure.

D. Performance Metrics

To evaluate the network performance different metrics are used including throughput, cache hit, delay, energy, cost, and offloading. Next, we briefly define the mentioned performance metrics.

1) Throughput: Throughput is a fairly accurate benchmark to evaluate network quality. It determines minimum average throughput per user to transfer data from the transmitter to the receiver successfully, which improves QoE. In other words, QoE is an acceptability of data service perceived by end-users. Hence, it is an end-to-end data service performance measurement from users' point of view depending on the throughput particularly in case of video streaming.

2) Delay: The delay of the system represents the total time taken from content request generation to content delivery. Network's delay is based on the request processing and transmitting data towards data requesting users.

3) Cache Hit: Cache hit is the probability of a user to obtain desired content from the local cache, either from its own or proximity users' cache, or serving BS's cache. Thus, cache hit is a helpful metric to judge the probability of obtaining desired contents from the cache resources. Hence, it provides information regarding handling of content requests and content delivery without accessing the core network.

4) Energy: The total network throughput to energy consumption defines energy efficiency. BSs and UEs are equipped with limited energy resources, therefore, placing contents at the network edge can provide energy conservation.

5) Offloading: Offloading is a technique to transfer the desired contents to the users via cache resources rather than through the core network. Data offloading can be categorized based on the level of synergy between unlicensed and cellular networks, and the participating BS and/or UE levels.

E. Challenges in Traditional Caching Mechanisms

We discuss different challenges pertaining to traditional caching mechanisms that need to be addressed for cache-enabled wireless networks. These challenges include mobility, big data handling, multi-tier caching, ubiquitous connectivity, and data redundancy.

1) Mobility: Mobility has serious implications on the performance of caching mechanism due to intermittent network connectivity. Data replication can be performed in various BSs to cater users' data demands. Since BSs can be equipped with limited cache resources, it causes degradation in network performance. If the network can observe the user mobility pattern accurately, requested contents can be cached in appropriate BSs or UEs according to the users' mobility path. This can maximize the cache hit rate and less contents will be fetched from the core network. However, it is a cumbersome task to accurately predict users' mobility. For mobility scenarios, effective and accurate prediction models need to be explored to enable efficient cache services.

2) Data Redundancy: Most of the existing works consider popularity based data caching. This causes wastage of the scarce cache resources due to data redundancy. Few works consider user preference based caching to store contents following users' tendency and avoid data duplication in cache. However, mobile users randomly roam over places. This may put a user in such a situation, where it may be surrounded by users with different preferences, in such case, the cache hit rate reduces and contents have to be obtained from the core network. This leads to rise in the backhaul overhead. Therefore, a robust system is required to meet user demands at any time irrespective of its location.

3) Big Data Handling: There is a tremendous growth in the data volume. It becomes difficult to extract features of contents and determine the popularity of each content. This requires highly sophisticated processing mechanism to optimize the caching procedure. The caching scheme should be able to accurately observe the social relationships among users, their content request patterns, mobility patterns, and new arriving contents. Hence, intelligent approaches are required for data caching of popular contents.

4) Multi-Tier Caching: BSs and UEs are equipped with limited cache resources. The advantages of caching at user and BS levels can be jointly utilized to pool together cache resources, which can overcome cache resources' limitation. However, there are several underlying challenges, which include centralized caching approach in the form of users' clustering for data placement, unavailability of accurate CSI causing failure in content delivery [73], designing caching mechanism taking dynamic content popularity into account [74], accurate decision making for content placement at each level [75]. Hence, to meet diverse data demands, exploiting multi-tier caching at BSs and UEs' levels is required.

5) Ubiquitous Connectivity: IoT is one of the future communication technologies that is adapting data caching [76]. The concept of IoT such as smart security, health monitoring equipment, and manufacturing systems require low delay within milliseconds. This creates a significant challenge with the objective to determine optimal caching policies. However, existing caching approaches are not able to meet the low delay requirements of IoT.

Summary: The BSs coverage is limited and connectivity between BSs and users is uncertain. This incurs data fetching from the core network to fulfil diverse users' data demands. Cooperative caching can be a way to minimize data fetching from the core network, as it enables data sharing among BSs [77], [78]. But users are mobile and they can frequently move across different cells. Hence, to optimize data placement at BSs it is crucial to accurately predict users' mobility patterns. In addition, dynamic channel conditions such as channel fading and interference severely affects the cache mechanisms' performance [79]. All of these factors increases the complexity of designing an efficient cache mechanism at the BS level. Moreover, data caching can also be carried out at the UE level for leveraging D2D communications. But users are selfish by nature and they do not want to consume their scarce sources for the sake of improving network performance. Smart caching mechanisms are required that should consider users' selfish behavior and intermittent connectivity between users to save scarce wireless spectrum, BS and UEs resources such as limited battery and cache spaces. Furthermore, it is essential to consider users' dynamic data preference profile based on the probability of data requests of each user while performing caching at BSs and/ or UEs. In comparison to the number of contents, the available cache resources are limited. Thus, it is imperative to consider data popularity profile, data size, data type, data ephemerality, cache capacity, users' mobility patterns, and network conditions for designing efficient caching mechanisms. AI appears to possess potential to optimize prediction of data popularity profile based on individual user data demands and mobility patterns. AI can be a solution to maximize the future wireless network performance by leveraging intelligent cache mechanisms.

III. OVERVIEW OF ARTIFICIAL INTELLIGENCE (AI)

The data caching in wireless networks can be carried out using mathematical tools including optimization, game theory, stochastic geometry, and artificial intelligence. Optimization is the one approach, which can solve caching problem through determining the closed-form solution or heuristic algorithm. Game theory can solve the caching problem by considering various players such as BSs or UEs, competing with each

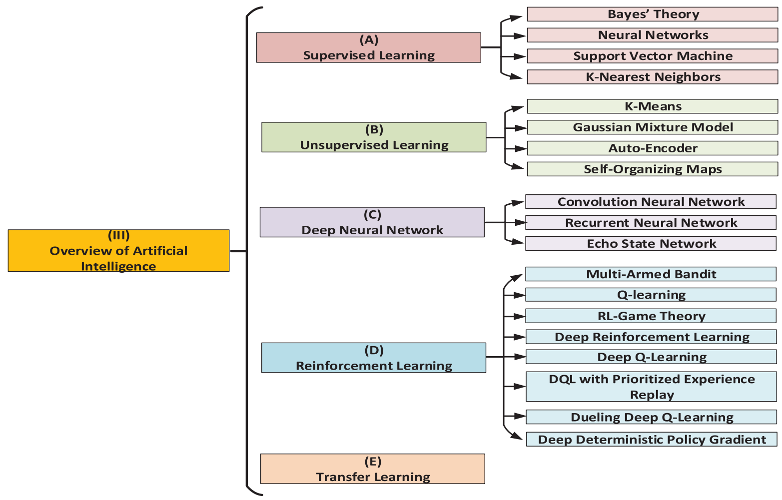

Fig. 3. Block diagram illustrating AI schemes in the overview of AI.

other to maximize the utility [80], [81]. In contrast to game theory and optimization, stochastic geometry models caching by devising statistical designing tool. With the evolution of AI-based decision making approaches, the decisions of content caching in wireless networks can be performed based on the learning process. AI has the potential to accurately predict data popularity profile utilizing users social behaviors, mobility patterns, and localization [82], [83].

AI is a powerful technique that is used for data mining to learn a system. Its main goal is to explore the relation between input and output for enabling the system to auto-process patterns of fed input data. The raw dataset is provided to the training process for decision making and provide accurate prediction. In a broader sense, intelligent learning approaches can be classified into supervised learning, unsupervised learning, RL, and TL. The Fig. 3 illustrates the hierarchical organization of the AI algorithms.

A. Supervised Learning

In supervised learning, a labeled data set is provided that contains inputs and known outputs for building or training a model based on the relations included in the dataset. Then, a new data set is collected and given to the learned model. This enables an algorithm to perform optimal predictions [84]–[87]. This approach is adopted in wireless networks for predicting user mobility [88]–[91], resource allocation [92]–[94], and handover optimization [95], [96]. The supervised learning can be further divided based on various applications. In the following subsections, we discuss supervised learning algorithms that are widely used in wireless networks including Bayes' theory, k-Nearest Neighbor (k-NN), Neural Network (NN), and Support Vector Machine (SVM).

1) Bayes' Theory: For statistical analysis the conditional probability axioms are followed by Bayes' theory. It helps to determine the probable occurrence of an event based on the prior information relevant to the event. This can be evaluated as

$$P(h|x) = \frac{P(x|h)P(h)}{P(x)}, \tag{1}$$

where h and x represent the hypothesis and evidence. $P(h|x)$ denotes the probability of h to be true given x, and $P(x|h)$ represents the likelihood of x on h [97]. The probability model is learned by Bayes' theory based on the training data. Evidence is the data sample and hypothesis is the class to assign for data sample. $P(h|x)$ denotes the belonging of the probability of a data sample to a certain class. $P(h)$, $P(x)$ and $P(x|h)$ are evaluated based on the training input dataset. $P(h|x)$ is updated for different classes during each iteration of training process. Akoush and Sameh in [98] devised Bayesian learning for predicting users' mobility patterns.

2) k-Nearest Neighbor (k-NN): This algorithm can be applied in situations where distribution of observations and results are unknown and classification is performed based on k-NNs. This algorithm first classifies a new sample based on the number of k-NNs of a particular class. Then, unclassified samples are classified in the same class [86], [99]. There are different approaches for distance measurement between unlabeled sample and their nearest neighbors such as Euclidean squared, Chebyshev, and City-block.

3) Neural Networks (NNs): NNs is developed based on the concept of human brain intelligence. The human brain contains the ability of performing highly complex, non-linear, and parallel computations. A NN is a collection of nodes regarded as neurons, which makes an algorithm powerful and efficient computational tool. NNs or Multi-Layer Perceptrons (MLPs) is a multi-layered hierarchical architecture that defines high level features by using level features. It can automatically abstract and draw features from input data for learning complicated data patterns [100], [101]. In a NN, nodes represent the components of neurons that perform all the nonlinear computations [102]. All nodes are linked by link weights simulating the connection of neurons of a human brain. A NN is composite of input, hidden, and output layers as shown in Fig. 4.

4) Support Vector Machine (SVM): Unlike NN, SVM takes structural behavior and provides better generalization through Structural Risk Minimization (SRM). SVM or large margin classifier maps input vector into a high dimensional feature space. Therefore, SVM is mainly used for pattern recognition

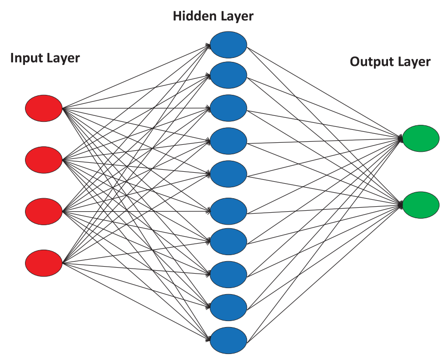

Fig. 4. An illustration of NN that is consist of three layers including input layer, hidden layer, and output layer.

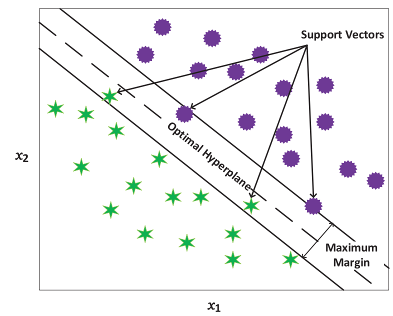

Fig. 5. An illustration of SVM model. Two classes are depicted with purple and green shapes. A dashed line for optimal separation hyperplane and two solid lines for margin hyperplanes in a binary classification.

and data classification. If input dataset is linear separable, a linear kernel is used that performs a crucial role in improving the accuracy of a classifier. SVM adopts linear or non-linear mapping for maximizing margin between different classes. It determines a separating hyperplane in feature space for achieving maximum margin between classes. The distance between classes is maximized by determining the largest margin between hardest points that ensures optimal decision region [97], [103]–[105]. There are three types of kernel functions including linear, Radial Based Function (RBF), and polynomial kernel. Fig. 5 shows SVM classifier utilizing linear mapping. Moreover, RBF or polynomial kernel can be user for non-linear mapping. The difference among them lies in terms of building hyperplane decision boundary between the classes, where linear and polynomial kernels provide low accuracy than RBF.

B. Unsupervised Learning

In unsupervised learning, a set of unlabeled input dataset is provided to the algorithm. The objective is to determine patterns and learn by clustering data into multiple groups based on the similarity. Unsupervised learning is most widely utilized in data collection and clustering problems [86]. The data clustering feature of unsupervised learning can be utilized in wireless networks for clustering users following their data request patterns [106], [107]. Afterwards, contents can be cached based on the prediction of users' data demands. In the following section, we discuss profound unsupervised learning algorithms.

1) K-Means: K-Means is a clustering algorithm. It assigns unlabeled input data to different clusters. To implement clustering based on the K-means algorithm, initial data set and desired quantity of clusters are required. For instance, to obtain a total of C clusters, the K-Means algorithm first initializes centroids of k clusters by randomly picking K nodes. Then, a distance function is utilized for labeling each node that exist near to the centroid. Each time a new node joins, assignment of new centroid is performed. This process continues until convergence [108]–[110].

2) Gaussian Mixture Model (GMM): K-Means performs hard classification and considers each cluster a spherical Gaussian. K-Means determines optimal mean for every Gaussian, while assuming each data point belonging to a single group. However, a data point may belong to more than one cluster. GMM provides soft classification by determining the probability of a data point belonging to multiple clusters. Moreover, GMM linearly superposes Gaussian distribution. It is worth emphasizing that the model parameters are the mixed coefficient of Gaussian component, mean, and covariance of all Gaussian distributions.

3) Auto-Encoder: Auto-encoder is a type of neural network to learn efficient coding and perform data compression or dimension minimization. The encoded information provides compressed representation of data set. The structure of an auto-encoder comprises of an input layer followed by several hidden layers, and finally an output layer. The input data is encoded through the hidden layers and the output layer reconstruct input layer. After training an auto-encoder, the decoder is removed and feature extraction is performed through the encoder. However, Auto-encoders have some limitations, which include input data distribution should be identical to the training data, need defining hyper-parameters that increase the complexity.

4) Self-Organizing Maps (SOMs): SOMs or Self-Organizing Feature Maps (SOFMs) monitor similarities in the input dataset and transforms an incoming signal of arbitrary dimension into two dimensional discrete map. It is useful in reducing the complexity by converting non-linearity of an input data into geometric relationships in a lower plane [111]. The number of neurons in a map layer is equal to the required number of clusters, and each neuron contains a weight vector and distinct location. The neurons weight vector is initialized in a map layer and data samples are picked from the input training dataset. Then, a distance function is utilized for measuring similarity between weight vectors and the data sample. In SOMs, neurons or units compete for input data. After feeding a sample, SOMs determines the neuron nearest to the current sample via calculating the weight between all the neurons and current sample. The weight is usually determined by the Euclidean distance and the wining node is referred as the Best Matching Unit (BMU) based on the fact that the highest similarity in the weight vector of neurons and samples that are allocated to the cluster.

C. Deep Neural Network (DNN)

DNN is used to perform complex computations and handle high-dimensional input data via multiple hidden

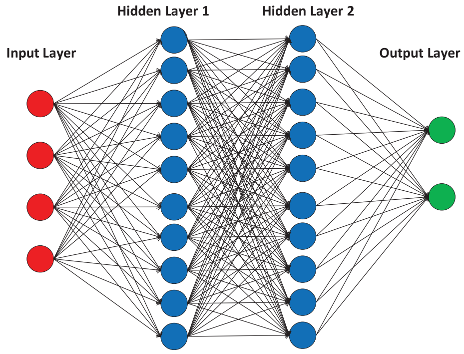

Fig. 6. An illustration of DNN.

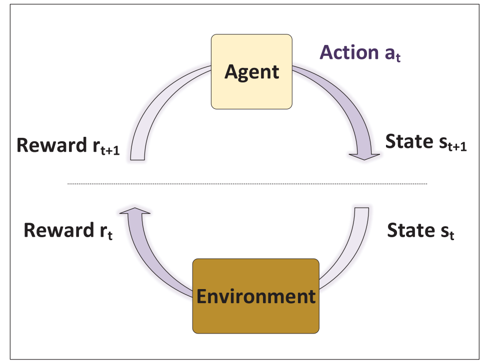

Fig. 7. General framing of Reinforcement Learning (RL) mechanism: An agent interact with environment and perform an action. The agent receives a reward or penalty based on the actions outcome.

layers [112], [113] as depicted in Fig. 6. It transforms input into output that either has linear or non-linear relationship. Each hidden layer's neurons based on the output of the previous layer perform feature training is regarded as a feature hierarchy. In contrast to RNN, DNN is a feed forward network where data flow occurs from input to output without complex feedback loops.

1) Convolutional Neural Network (CNN): CNN or shift invariant network animate biological mechanism where the neurons interconnections correspond to visual cortex of the brain. It requires less pre-processing by devising MLPs. CNN has three characteristics including shared weights, spatial-temporal sampling, and local receptive fields. Convolutional layers are major components of CNN for feature extraction. Similar to DNN, a feed-forward approach is also utilized by CNN [114], [115]. The CNN is efficient in reducing complexity by local sparse connections and sharing weights. This helps to minimize training parameters and perform spooling to reduce feature size.

2) Recurrent Neural Network (RNN): RNN shares same parameters at each layer and leverages sequential knowledge, which is different from DNN. At each time step, the same task is performed for each sequential element with variable inputs. This approach provides significant reduction in training parameters. RNN has a short-term memory. Its memory can be extended by utilizing Long Short-Term Memory (LSTM) that enables RNN to remember inputs for a long time [116], [117].

a) Echo state network (ESN): ESN is a type of RNN that provides efficient computation models and approximates nonlinear dynamic networks. A large number of neurons are devised in ESN, where all the connections between neurons are established in a random manner. Unlike RNNs, link weights of recurrent layer do not need training. Only link weights to output layer are required to be optimized that substantially simplifies training procedure. Therefore, error function becomes quadratic, which can be differentiated to a linear system.

D. Reinforcement Learning (RL)

RL is inspired by behavioral psychology. The learning is done in response of stimuli, and actions are performed based on punishment and rewards. It enables an agent to find optimal actions in order to optimize performance. It maximizes the reward instead of determining latent structure. In RL, an agent performs an action and obtains immediate rewards or cumulative rewards in return to maximize system performance in the long run as shown in the Fig. 7. RL has two modes of operations. An agent can either exploit an action that provided high rewards previously or explore new actions to further raise rewards. Hence, a balance between two is necessary for optimal performance [118].

1) Multi-Armed Bandit (MAB): MAB consists of an agent and independent arms. An arm is picked to perform an action during each time interval and in return a reward is obtained. An arm is played with an expected reward following Independent Identical Distribution (i.i.d.). The objective is to determine which arm should be played during each time interval to maximize long term expected reward. In MAB, there is a need to perform tradeoff between exploration and exploitation. An exploration of new arms is performed to estimate mean rewards, conversely, exploitation of known arms can be performed that provided high rewards in previous time intervals. Agent should be made aware of each arm reward, which enables playing arms with high expected reward. A regret phenomenon is induced to measure MAB performance. Regret defines difference between obtained reward and optimal reward. A variant of MAB is Combinatorial MAB (CMAB), which plays a set of arms (super arm) during each time interval, where reward of each super arm depends on the underlying arms individual rewards.

2) Q-Learning: Q-learning is a model-free learning mechanism that determines optimal policy to perform action in each state. Q-learning based algorithm constitutes of a set of states and actions. In every state, an action is performed, and a reward is obtained as the result. Moreover, a Q-table is maintained that contains reward values for each state-action pair. The Q-table facilitates agent to perform an action in the current state, yielding a high reward. According to the obtained rewards, agent keeps updating the policy.

3) RL-Based Game Theory: RL-based game theory concentrates on the interactions among decision makers. The idea of game theory involves set of players, strategies, and utility functions. The decision makers function as players that use utility functions for selection of optimal policies. The

TABLE III
ADVANTAGES & DISADVANTAGES OF THE ARTIFICIAL INTELLIGENCE (AI) ALGORITHMS

Algorithm	Problem	Advantages	Disadvantages
Bayes' theory	Classification	• Implementation is easy • Accuracy and computational efficiency is high	• Accuracy is affected by less data • Hard to handle large data
k-NN	Regression, classification	• Negligible cost of training • Robustness in case of noisy data • Tolerant in selecting distance functions	• Time consumption is high to determine neighbors in large data set • Vulnerable to noise data
NNs	Regression, classification	• Easy implementation • Training is not necessary • Efficiently deals high dimensional data	• Processing time is proportional to neural network due to tuning large number of parameters • Hard to determine number of layers and neurons
SVM	Regression, classification	• Accuracy is high • Suitable for both linear and non-linear separable input data	• Computationally expensive • Require large memory • Overfitting issues in case of noisy data
K-Means	Segmentation	• Easy implementation • Efficient in dealing large dataset • Automatically assigning objects to clusters	• Hard to determine number of clusters • Random selection of initial centroids • Sensitivity towards outliers
GMM	Classification	• Cluster covariance flexibility • Classify non-temporal behaviors	• Need to define quantity of mixture models • Computations are expensive
SOMs	Segmentation	• Easy data mapping for prediction • Efficient in handling high-dimensional dataset	• Lacking parallelization of large dataset • High computation cost
DNN	Feature extraction	• Enable massive parallel computing • Robust to data variations	• Large amount of data is required • Requires information about topology, parameters, and training technique
CNN	Feature extraction	• High accuracy • Automatic detection	• High dimensional cost • Require large training data
RNN	Sequence prediction	• Sequence data modeling • Computationally efficient	• Training is hard • Hard to process long sequences
ESN	Handle disordered time series	• Training is easy and fast • Low computational cost	• More hidden units are required
MAB	Action selection based on feedback	• No delay effect • Error correction during process	• Cost of arm switching • Large time horizon required
Q-learning	Sequential decision making	• No data set required • Sustain changes	• Balance of exploitation and exploration • Dimensionality curse
DRL	Complex target function	• High performance • Handling large state and action spaces	• Computationally expensive • Need large data to achieve convergence
TL	Prediction	• Save training time	• Non-adaptive to changes

game theory has two divisions including cooperative games and non-cooperative games. In the cooperative game theory, there is a cooperation among decision makers (players) and establish multiple coalitions. Optimal policies are selected for maximizing the utility of coalitions. Conversely, for maximizing individual utility, non-cooperative games enable players to compete with each other and pick policies in an independent manner [119]–[121].

4) Deep Reinforcement Learning (DRL): RL has the potential to exploit strengths of both the supervised and unsupervised learning designs. In RL, rewards are devised to enable agent to behave in an appropriate manner to optimize the system performance. The most well-known RL type is Q-learning due to its simple architecture but it experiences curse of dimensionality with increasing state and action spaces. Therefore, DRL is designed to handle large dimensional spaces by combining RL and NN. Recently, Deep Q networks are designed to combine Q-learning with DNN to address overestimation problem of Q-learning techniques [122].

a) Deep Q-learning (DQL): DQL implements Deep Q-Network (DQN) to determine approximation value of $Q(state, action)$. DQL employs experience replay and target Q-network. First, replay memory is initialized with system transitions or experiences. Then, experiences or batches of transitions are selected from the replay memory to train the DNN that infers Q-values in return. However, during the training process Q-values are shifted, which makes value estimation hard that may lead to instability. Therefore, target Q-network is employed to keep updating the primary values of Q-networks, which reduces correlation between target and estimated Q-values to achieve stability.

b) DQL with prioritized experience replay (DQL-PER): Experience replay technique enables reuse of experiences. However, replay transitions are performed at the same frequency according to the experiences of agent without considering significance of replay transitions. Therefore, prioritizing replay transitions based on their significance can improve the efficiency. For this reason, DQL-PER is designed to perform probabilistic sampling of transitions related to an absolute error. In this manner, new transitions with highest probability are stored in the replay memory.

c) Dueling deep Q-learning (Dueling DQL): The Q-learning estimates both the state-value and action-value functions. However, it is not always necessary to estimate both the values at the same time based on the system dynamics. Inspired by this idea, Dueling Deep Q-Learning approach is designed that uses two sequences of fully connected layers for DQN rather than a single sequence of fully connected layers. This helps to estimate state-value function and action-value function separately. Then, at the end both the sequences are combined to produce one output $Q(state, action)$.

d) Deep deterministic policy gradient (DDPG): The DQL has the potential to solve high-dimensional state spaces and discrete action spaces. However, DQL cannot be applied on systems having continuous actions because they select best actions that can optimize their Q-value functions. Therefore, DDPG has been designed to learn competitive policies by exploiting low dimensional observations. DDPG utilizes actor and critic networks to evaluate the target values, where the weights of target network are updated by learning network through slow tracking. Hence, target values vary slowly and optimizes the learning stability.

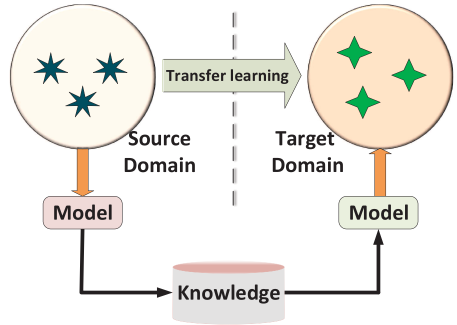

Fig. 8. An illustration of TL method for knowledge transfer from source domain to target domain.

E. Transfer Learning (TL)

TL is a mechanism of learning from one problem and applying the gained knowledge or information to other related problem. An illustration of TL approach is demonstrated in Fig. 8. TL helps to enhance the learning process by transferring knowledge between source and target domains. Hence, TL is beneficial when training dataset is insufficient to train network from scratch. TL can be devised in regression, clustering, and classification problems. Despite the benefit, this approach may show low performance if the relationship between source and target domains is inadequate.

Summary: In this section, we discussed the fundamental concepts of supervised learning, unsupervised learning, RL, and TL. Furthermore, we also discussed extensions of all the mentioned approaches. The supervised learning schemes are used for classification and clustering, where unsupervised and RL are used to achieve clustering and making smart decisions, respectively. Moreover, TL circumvents the need of training at every time. A model trained for one environment can be utilized to handle the new but similar environment. This speeds up the training process and improves the accuracy and effectiveness of the learning mechanism. In Table III, we provide comparison of state-of-the-art AI algorithms.

IV. AI Based Caching Approaches

The performance of a caching algorithm is based on the accurate information about data popularity and users' preferences, which require a long period of observation. It is also important to monitor temporal variations in data popularity and their demands as it plays vital role in designing an efficient data caching policy. Despite intrinsic complexity and challenges, AI based caching mechanisms seems to be a promising way to improve the accuracy of prediction systems. In this section, we categorize the AI based caching research works considering network performance in terms of data offloading, cache hits, energy, delay, cost, and throughput. Fig. 9 provides organization of all AI based caching approaches. The implication of AI for data caching in wireless networks is shown in Fig. 10, which depicts information sharing by exploiting TL between networks that are using RL to optimize network performance.

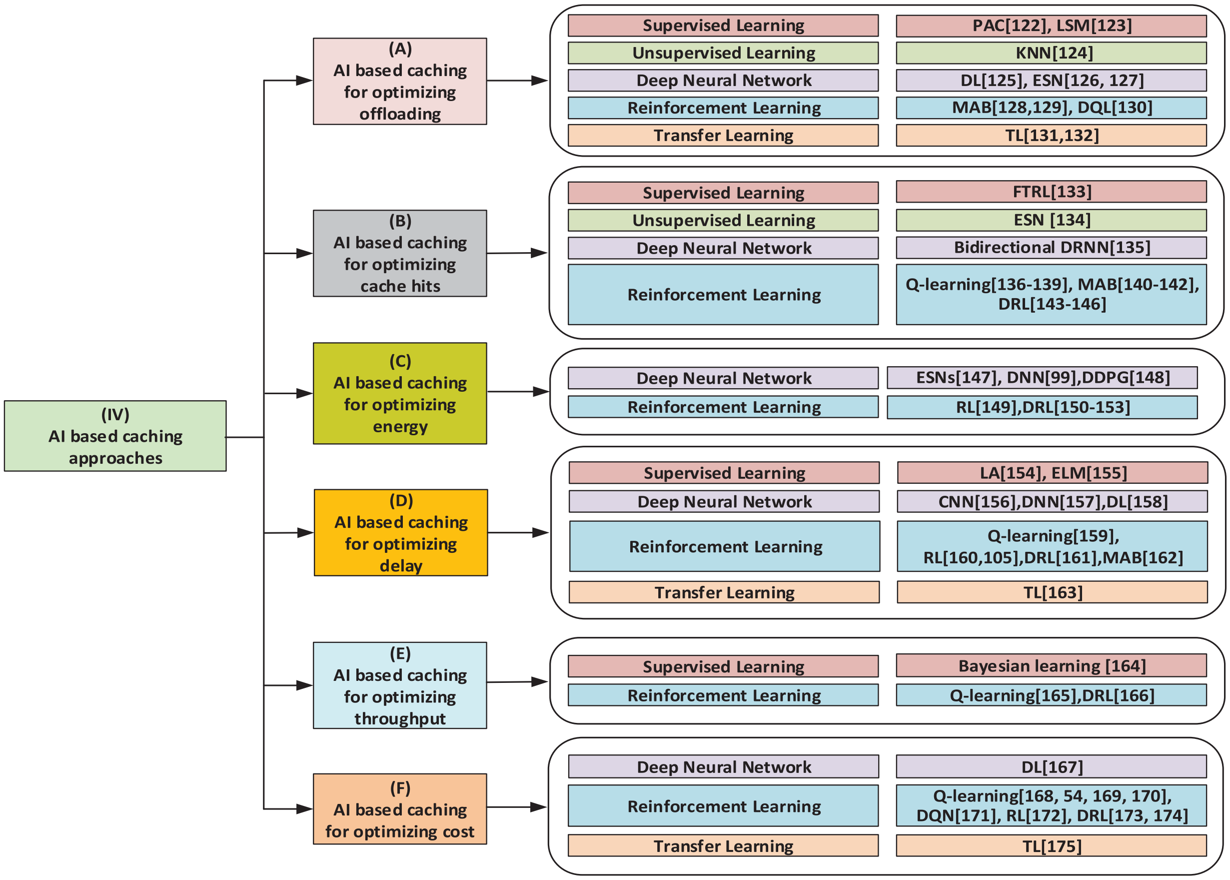

Fig. 9. Organization of all AI based caching approaches.

A. AI Based Caching for Optimizing Offloading

In this subsection, we discuss several AI based caching algorithms to optimize data offloading in wireless networks. Data offloading through the cache resources refers to fulfill users data demands through the cached contents in order to alleviate traffic burden over the cellular link. AI based caching approaches to optimize data offloading are divided based on their respective AI techniques such as supervised learning, unsupervised learning, DNN, RL, and TL.

1) Supervised Learning: However, the popularity of contents vary with respect to time. This may cause unavailability of demanded contents leading to failure of caching mechanism. This problem is handled by Bharath *et al.* [123] by considering evolving popularity of contents with respect to time. The unavailability of requested contents in the SBSs is measured by devising a cost function to capture offloading loss, and Probably Approximately Correct (PAC) is derived for investigating high probability bond on different offloading loss. It helps to determine the data offloading difference between the estimated and optimum value. The BS computes estimation of Rademacher complexity and its divergence based on the known data request at any given time. This refreshes contents in the cache resources if divergence of data profile is large than a certain threshold.

UAVs can be used as an aerial base station to provide Line-of-Sight (LoS) link for the users to optimize data rate and delay of wireless networks. Moreover, UAVs can also be used as aerial relays between source and destination to provide a LoS link. There are several challenges in deploying UAVs regarding channel modeling, energy efficiency, link planning, and resource allocation. UAVs can be equipped with LTE to exploit both the licensed and unlicensed spectrum for providing data services to the users. LTE-U enables UAVs to utilize both the WiFi band and unlicensed spectrum to satisfy users data rate requirements. It appears to be a promising approach to exploit data caching and LTE-U resource allocation at UAVs to extend the capacity of UAVs based data service. Since spectrum allocation and data caching is dependent on each user's association and request frequency and UAVs may not determine the users' data demands. Consequently, the problem becomes intractable and conventional algorithms cannot provide optimal solutions. Therefore, Chen *et al.* [124] proposed Liquid State Machine (LSM) algorithm to improve the accuracy of predictions of data requests distribution and optimal resource allocation. The authors considered a wireless network consisting of UAVs and WiFi Access Points (WAPs) to provide data services to the users as depicted in Fig. 11. A user can obtain desired content from UAV and if the desired content is not cached by the UAV, then UAV will fetch content from the server and forward the content to the requesting user. The proposed LSM approach predicts users' data demands distribution with limited information regarding network conditions in order to avoid data fetching from the core network. LSM

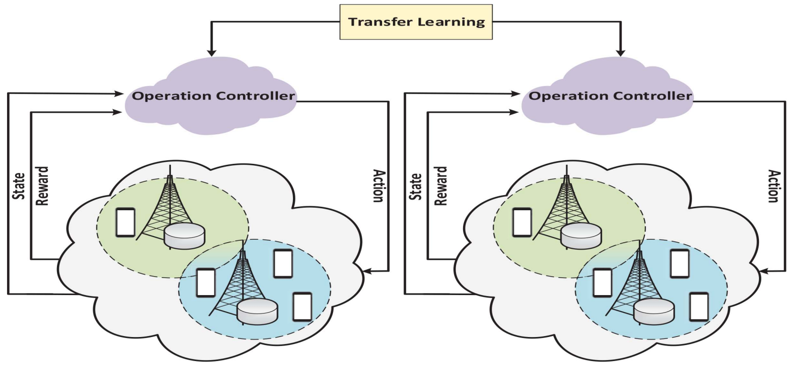

Fig. 10. An illustration of implementation of AI in cache-enabled wireless networks. Wireless networks perform data caching by using RL approach to optimize data sharing among users. Moreover, the TL is exploited to share caching knowledge between the networks to reduce caching mechanism's complexity and increase its efficiency.

store user behavior information and utilizes the behavioral patterns to predict data demands distribution and autonomously perform appropriate resource allocation following network dynamics. The proposed approach enables UAVs to automatically decide resource allocation policies for maximizing UEs stable queues based on network conditions and meet users queue stability (i.e., metric to measure data transmission rate to achieve dynamicity of data transmission) requirements. Moreover, based on users' association and data demands distribution contents are cached at the UAV, and resources are allocated autonomously by exploiting limited information regarding network dynamics.

2) Unsupervised Learning: The spatio-temporal request patterns and new arriving contents poses severe impacts on the performance of the caching mechanisms. In [125], Shen *et al.* performed data caching in Ultra Dense Networks (UDNs) with the objective to offload data traffic from the backhaul by designing an optimal cache mechanism. However, it is a cumbersome task due to random content requests. The authors utilize both *k*-means and *k*-NN algorithms for determining the hidden spatio-temporal content request pattern for both inter and intra-cluster caching and classification of new contents, respectively. The proposed approach has low complexity and high accuracy that significantly reduces backhaul data traffic congestion. The cached data can be shared among users via D2D communications. Hence, social features of users should be used for secure and efficient communication.

3) Deep Neural Network: Designing proactive caching mechanism is challenging because content popularity matrix has sparse information regarding correlation between the content and specific user. Moreover, content popularity matrix does not provide information about latent patterns for interaction between users and contents. DL is a promising solution that avoids the needs of data labeling for the training purposes. Rathore *et al.* [126] proposed DL based proactive data caching scheme in Cellular Networks (DeepCachNet). The proposed scheme collects information from large number of connected users. Auto-encoder and stacked denoising auto-encoders are devised for users and data feature extraction, respectively. The user's features provide information regarding user demography, gender, age, social group, and data interests. Data feature extraction requires processing of auxiliary data descriptions of the raw data. User-content pairing is performed to define the rating or popularity of each content for individual user. Then, based on the processed information, data caching is performed at SBS level to offload data traffic through the cache resources for alleviating data traffic from the backhaul. The proposed approach is efficient to deal with data sparsity in content popularity matrix that poses limitation on determining correlation between content demands and user interaction, which provides substantial gains in proactive caching.

The wireless network entities are equipped with limited cache resources, therefore, there is a dire need to optimize data caching for maximizing data offloading. Chen *et al.* [127] proposed data caching at Radio Resource Heads (RRHs) in Cloud Radio Access Network (C-RAN) that are controlled by Base Band Units (BBUs). The distribution of content requests is predicted by BBUs and based on the prediction RRHs are clustered. ESN along with sublinear algorithms are utilized that is a RNN with sparsely connected latent layers. It optimizes joint data caching and RRHs clustering despite limited information regarding network and users' states. First, the ESN approach predicts data request distribution and accordingly contents are placed in RRHs based on the data popularity profile determined by the requests of associated users during each time slot. A sublinear algorithm is devised to achieve an optimal data placement for maximizing data offloading. The proposed algorithm jointly regulates data caching and RRHs clustering based on individual user data preferences. Finally, contents are also placed in BBUs to avoid data fetching from the core network that infers traffic burden on the system.

In [128], Chen *et al.* provided extension towards more complicated and practical mobility scenario, where all the mobile users perform data caching and thus there can be severe complexity in designing a caching mechanism. To tackle varying network environment, the authors derived ESN's memory capacity for periodic input to capture each user's mobility pattern with low complexity. A sublinear algorithm is devised for data caching by utilizing samples from the content requested distribution. Subsequently, contents are placed in each cluster of RRHs. The data placement is performed following the number of requests generated by associated users during each time slot, which continuously varies due to users' mobility.

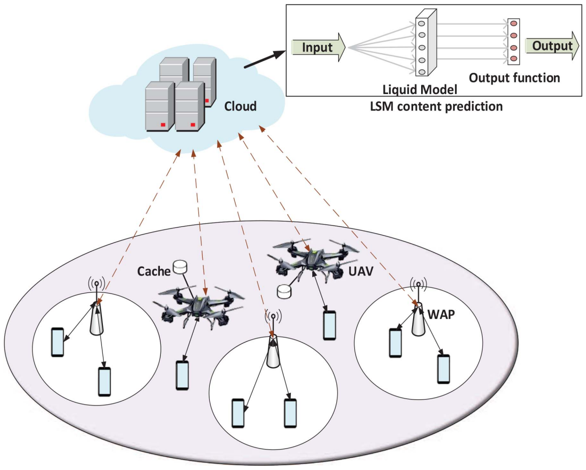

Fig. 11. An architecture illustrating Heterogeneous wireless network with cache-enabled UAVs.

4) Reinforcement Learning: The nature of data popularity profile is dynamic with respect to time, it evolves with passing time. Blasco and Gündüz in [129] proposed data caching in a SBS central controller to offload data traffic during peak hours through cache resources. To determine unknown popularity of contents, central controller can perform refreshing of cache resources after regular intervals of time to optimize performance of limited cache resources. A RL-based MAB scheme is utilized, where agents lack system information and repeatedly perform action exploiting current information during each stage in order to maximize accumulated reward. In contrast to classical MAB, Combinatorial MAB (CMAB) is proposed for exploiting more than one action to maximize reward. The authors used Combinatorial Upper Confidence Bounds (CUCB), Modified CUCB (MCUCB) and ϵ-greedy algorithms for determining data popularity profile and achieve optimal data caching to offload more data traffic via cache resources. The ϵ-greedy algorithm showed high performance in terms of various system parameters such as data profile, an increasing number of users, limited cache size, and a large number of contents. Blasco and Gündüz in [130] considered a Wireless Infostation Network (WIN) that models a circumscribed range wireless network containing substantial cache capability. It supports content-level selective data offloading by providing high data rate service to the users under coverage via broadband links. This work considers data caching problem as a CMAB problem with switching cost.

When state dimensions increase, then traditional RL approaches experience curse of dimensions. Therefore, DRL approach can be devised to optimize learning procedure. In [131], the authors focused on security of Mobile Social Networks (MSNs) by exploiting social trust feature derived via Bayesian interference and Dempster-Shafer theory. DQL is devised for optimal decision making in terms of autonomous allocation of network resources for data caching and computation. All of the parameters are sent to the DNN that provide optimal actions. It helps agent to train and update DNN based on the rewards.

5) Transfer Learning: The data demands are based on the popularity of contents, which is unknown and dynamic. For accurate placement of contents near to users, it is essential to accurately determine evolving data popularity profile and users' data access patterns. The tracking of dynamic data popularity profile with respect to time is difficult, which makes it hard to train the cache mechanism by using a large dataset. Alternative solution is utilizing known information of one network to train other similar network. This can reduce complexity and training time significantly. In [132], Nagaraja and Nagananda considered data caching at SBSs in a Heterogeneous Network (HetNet) to meet data demands of a large number of users distributed following PPP. The contents are placed in the cache of each SBS to enable the users to obtain its desired contents from the closest SBS. This provision enables users to meet their data demands through SBSs rather than via the core network, incurring low burden on the system. The data offloading loss is estimated through a cost function due to uncertainty about data popularity profile. The data popularity profile is determined by utilizing TL approach for computing empirical cost function in a centralized manner to monitor evolving data requests. Therefore, samples from source domain such as social network are used for improving data popularity estimation accuracy in the target domain. The results show improvement in the estimation of data popularity profile and data offloading, thus, manifests an achievement of high accuracy in the estimation process.

The challenges of how, when, and where to cache contents considering limited cache resources is tackled in [133] by Baştuğ *et al.* The authors proposed TL based data caching in SBSs for extracting users' latent contextual information (e.g., data request history, social ties). Samples from source domain are used to optimize data caching in the target domain, which is subject to the constraint on cache capacity and backhaul capacity. The proposed mechanism show its benefit in handling data sparsity and learning from scratch issues.

B. AI Based Caching for Optimizing Cache Hits

The cache mechanism efficiency depends on the efficient utilization of limited cache resources. For high performance gains it is crucial to avoid redundancy and cache diverse contents that may lead to high cache hits. A high cache hits means more data requests are fulfilled through the cache resources. This requires to adapt intelligent caching approaches for maximizing cache access. Subsequently, we mention several AI based caching approaches that focus on optimizing cache hits.

1) Supervised Learning: The network performance degradation can be halted by learning user data request patterns to determine future data popularity profile. Jiang *et al.* in [134] performed user preference based caching and focused on an online content popularity prediction scheme. This scheme utilizes independent user preferences and data characteristics to learn offline user preferences. Follow the Regularized Leader-Proximal algorithm and Online Gradient Descent (OGD) technique are devised, which do not require information regarding contents popularity distribution and continuous offline training. The proposed approach provides fast future prediction of content popularity with less complexity in an online manner. Furthermore, a track of variation in popularity of contents over time is managed without incurring any delay. This helps to optimize data caching for increasing the cache hit rate. However, there is a long queue of data requests that raises a control policy challenge.

2) Unsupervised Learning: In Content Centric Networks (CCNs) the prime focus is on content instead of the location of placed content for reducing data retrieval time. But multiple users associated with a single cache node may demand distinct contents, which can degrade cache hit because cache might not be able to fulfill diverse demands. Yin *et al.* in [135] improved the performance of CNNs by anticipating popularity distribution of contents via ESN and contents are placed in the cache resources of SBSs. ESN is a special type of Recurrent NN with additional dynamic reservoir. In general, the ESN system model consists of three layers: input layer, hidden layer, and output layer. Three layers are connected by input weight matrix and output weight matrix. In addition, the nodes of hidden layer are connected by the hidden layer matrix. In the training stage, only the output weight matrix needs to be changed making the ESN training process simple and efficient. Due to the time-varying characteristics of dynamic systems, ESN is much more suitable for handling the problem of the dynamic system modeling such as prediction. ESN can predict and cache contents optimally. Taking regularity of users content request, ESN can establish the relationship between user information and requested contents for accurate prediction. ESN requires a large training dataset, which increases the training time and computational complexity.

3) Deep Neural Network: Proactive data caching has the potential to reduce data access delay and overcome burden from the backhaul link. However, proactive caching requires accurate prediction of data popularity that is unavailable and evolve with respect to time. To address the aforementioned challenge, Ale *et al.* [136] performed learning of hidden data demands association and predicted evolving data demands with respect to time. The authors proposed an online proactive data caching policy based on Bidirectional Deep RNN (BRNN) for predicting data demands and cache status. On the first layer, 1-D CNN is exploited to handle data sparsity and minimized the computing costs by lowering dimensions of input dataset. Then, BRNN is devised for predicting evolving users' data demands. In contrast to NN that does not maintain history of previously learned information, BRNN performs block passing of the learned information in both forward and backward directions to enable the model utilize both the predecessor and successor knowledge. Finally, Fully Connected CNN (FCNN) is devised to perform learning and sample prediction from the BRNN.

4) Reinforcement Learning: For increasing the percentage of satisfied users via cache resources we have to reduce cache misses, which in turn reduces the backhaul traffic. Hence, the volume of reduction in network congestion is proportional to the cache hit rate. This requires ingenious data caching techniques to improve cache hits. Zhang *et al.* in [137] proposed Grouped Linear Model (GLM) to determine the future data requests distribution based on previous requests. There is randomness in the popularity of contents due to generation of new contents with every passing time, and each user has an individual content request pattern. A non-stationary MDP modeling is performed to perceive dynamic nature of content popularity profile with the objective of maximizing the cache hits. Then, RL-based Model-Free Acceleration (RLMA) approach is devised to perform data replacement by jointly considering cache hit rate and data replacement cost. Thus, learning process is accelerated in a dynamic environment by updating Q-value with imaginary samples.

Wireless Content Delivery Networks (WCDNs) is getting enormous attention from researchers due to its potential to minimize network congestion proportional to cache hit rate. Hence, in WCDNs prime objective is improving cache hit rate by distributed servers by updating cached contents. Cooperation among distributed servers may provide rise in cache hits. Sung *et al.* in [138] exploited cooperation among servers in WCDNs for improving cache hits while taking the cache capacity constraint into account. The authors utilized reactive caching paradigm for this purpose to deal with evolving content popularity. Based on users' different content preferences Q-learning is used for achieving an optimal data caching. First, content replacement problem is modeled as MDP. Then, a multi-agent Q-learning technique is used. The critical part in designing a Q-learning-based algorithm is to define a reward function that governs the algorithm's performance. To achieve a high reward in terms of the cache hit rate, it is essential to perform an optimal content replacement action. When cache new contents, there will be a rise in cache hits, leading to performance gain: otherwise, performance loss will be caused by cache misses. The proposed mechanism outperforms most widely used state-of-the-art schemes including LRU and LFU caching mechanisms. The potential of RL is also exploited by Shahriari *et al.* [139]. The authors proposed Generic Online Learning (GOL) scheme based on RL with hierarchical structure for 5G C-RAN to achieve load balancing. It creates abstract features in time that is adaptive to the optimal solutions. This self-organizing hierarchical state-action space system can interact with dynamic environment based on the input and output data of the network. In return a set of delayed feedback is obtained from the environment for adjusting internal structure. The results show that GOL is an efficient mechanism to reduce cache hit misses.

Despite the advantages of Q-learning in a multi-agent distributed systems, it does not guarantee optimal convergence. Lin *et al.* [140] performed distributed data caching by exploiting Multi-agent Reinforcement Learning (MARL). MARL exploited Joint Action Learners (JALs) and Independent Learners (ILs). JALs in contrast to ILs can observe other agents' actions. The authors considered each SBS as an IL and can provide contents to the users under its coverage. Each SBS optimizes distributed caching to maximize network performance in terms of improving cache hits. First, cooperative repeated game is used to model distributed caching, then MARL is exploited to increase cache hits. Further, Frequency maximum Q-value and distributed Q-learning algorithms are designed to improve the performance of distributed data caching policy. The results depicted that the proposed scheme is better than the Q-learning scheme.

Müller *et al.* [141] used RL to perform smart caching in SBSs for maximizing cache hits by exploiting distribution of contextual information of data. An online learning named as Contextual learning with the uniform partition algorithm is proposed. This learning is based on Contextual Multi-Armed Bandits, which performs data placement following varying data popularity profile. The proposed algorithm uniformly partitions the context space into smaller sets. Then, each SBS learns the expected data demands independently in each of the sets based on users' demands. Multiple actions of selecting multiple contents at any given time are performed based on the context-dependent data demands over complete timeline. In another work [142], a Contextual Multi-Arm Bandit scheme is used to achieve content-aware proactive caching. This increases the probability of data access through cache resources. The variation in content popularity needs to be determined in an online manner by observing context information of all the connected mobile users. According to the context-awareness, contents are cached for increasing the probability of achieving cache hits. The proposed algorithm calculates service differentiation by prioritizing users' groups while cache entity contains no prior knowledge regarding data demands. This requires a tradeoff between exploration and exploitation to cache new arriving contents. It may benefit cache hits by exploiting available knowledge of contents popularity (e.g., contents with high requests). The results show a significant performance gain in result of utilizing contextual information for proactive data caching to maximize the cache hit rate. Altervative solution is to design an autonomous system to keep track of traffic and data popularity profile variations In [143], the authors exploited distributed caching in a wireless network via RL. A coded caching is devised to learn the data popularity profile through CMAB scheme. Contents are coded by fountain coding and segments of data are proactively cached at SBSs to meet users' data demands. The instantaneous reward for each user is employed, in order to reduce replication of data placement in cache resources and thus benefits cache hits.

The integration of RL and deep learning has strengthened the scope of solving highly complex and challenging data caching problems. Its benefits can be utilized in dynamic network scenarios, where the knowledge of network state is lacking and there is a large delay in the feedback. Inspired by the DRL to solve complex problems, to solve data exploration and caching problem Zhong *et al.* in [144] utilized Wolpertinger architecture along with DRL to optimize data caching at BSs. The authors considered no prior knowledge about probability distribution of contents. The benefit provided by the Wolpertinger architecture is limiting size of action-space. Deep deterministic policy gradient is utilized for training purpose. The proposed model contains actor network, k-NN, and critic network. First, an actor network takes cache state and the current content request as its input, and provides a single proto actor at its output. Then, k-NN receives the single actor as its input, and calculates the distance between every valid action and the proto actor in order to expand the proto-actor to an action space, where each element is a possible action. The critic network takes the action space as an input and refines the actor network based on the Q value. DDPG is applied to update both critic and actor networks. The proposed approach shows better performance than First-In First-Out (FIFO), LFU, and LRU.

Enormous ephemeral data is produced in IoT. The transmission of such a large amount of data may cause network congestion and delays. This can be circumvented by deploying cache resources at the network edge. However, it requires joint consideration of data ephemerality and varying context, which increases the complexity. A better solution is the coupling of RL and NN for improving the predictions using users contexts, network conditions, new arriving contents, and users' mobility. Zhu *et al.* in [145] proposed edge data caching in IoT while taking data refreshing into account. A cost function is used for achieving tradeoff between data placement and cost of data communication. The cache placement problem is modeled via MDP to reduce the long term cost. If data is obtained from the data generator, then freshness loss cost is low but cost of communication is high. Conversely, when data is obtained from cache resources, the freshness loss cost is non-negative but there is a low data communication cost. A policy is designed to select actions for minimizing cost of fetching every data. The data item cache placement problem is modeled by MDP. The data consists of time-stamp and lifetime fields. The time-stamp field data generation time and lifetime denote the data validity time. DRL is used for caching IoT data without prior knowledge of data popularity profile, data request patterns, and contextual information. DRL predicts the system state and autonomously adopt caching scheme based on the previous data and existing observations. The cache resource performs cache update for refreshing cached contents in order to replace old data items with the new one.

Non-orthogonal Multiple Access (NOMA) can transmit a combination of multiple contents towards users and objective is to allocate appropriate power for each content signal. NOMA is a promising approach to enhance network capacity and users' QoE. In NOMA multiple data signals are superposed with different power levels, where a common radio spectrum of frequency and time is shared by all the users. Then, Successive Interference Cancellation (SIC) is employed by all the data receivers for successful data decoding after decoding a sequence of received signals before obtaining the desired signals. But in case of a cache-enable wireless network where NOMA is employed, it increases the challenges as in addition to channel condition there is a need to perform accurate data requests prediction. Therefore, it necessitates to explore design a power allocation strategy along with achieving fairness among users. Further, cached contents should be exploited to cancel interference in order to maximize achievable data rate. Doan *et al.* [146] exploited advantages of Non-Orthogonal Multiple Access (NOMA) while performing data caching at both the SBS and UEs level in order to maximize cache hits. Moreover, data service fairness and probability of successful data decoding is increased to optimize QoS. First, divide-and conquer method is proposed to achieve an optimal resource allocation. Since there are randomized channel gains, and rewards also exhibit random behavior, hence, DRL is devised to enable bandwidth sharing for all the users while avoiding noise in the training dataset. In the beginning exploration is performed based on four steps: data demands, power allocation vector, channel coefficients, and computation of obtained rewards. Exploration infers a list of states and corresponding appropriate actions to optimize the reward. Subsequently, a parametric function is trained to perform optimal action. Afterwards, exploitation of trained model is performed to improve the power allocation and data caching.

The dynamics of content popularity are based on time and homophily that can be predicted by observing evolving users' data demands with respect to time. Further, data preferences of one user does not depict the other user data priorities. Jiang *et al.* [147] exploited DQL to determine the user preferences and data popularity features to maximize cache hits. First, user preference prediction and content popularity prediction algorithms are utilized. Then, DQL is devised to optimize data caching at the UEs, Fog access points, and BSs. Finally, cached content update policy is utilized to perform accurate prediction of user data preferences and data popularity. The results demonstrated a substantial growth in cache hits.

C. AI-Based Caching for Optimizing Energy

Modern wireless technologies (e.g., ubiquitous communication, online gaming, video streaming, etc.) bring a high cost of energy consumption. Data caching can also reduce energy consumption of the wireless networks. How to implement caching technique for optimizing energy conservation is a serious research question.

1) Deep Neural Network: User exhibit mobility and to provide data services to the mobile users it is imperative to utilize mobility patterns of users to optimize data caching. It is difficult to utilize heuristic methods for making online decision, since a large number of iterations will degrade the computation process. To deal with large number of features DRL appears to be a promising approach to optimize decision making. Inspired by this concept, Chen *et al.* [148] focused on data caching in UAVs to optimize users QoE. The contextual information of users such as job, gender, data request pattern, and mobility pattern is utilized. After exploiting users' contextual information, association between users and UAV is determined for performing data caching at each UAV according to the distribution of data request predictions. A conceptor-based ESN is utilized for efficient prediction of user requests and mobility of users under limited information regarding network and user states. The proposed algorithm enables ESNs to separate the users behavior into several patterns and learn these patterns with various non-linear systems. The delay is minimized and user experience is improved by determining the optimal position of each UAV and data placement by exploiting human-in-the-loop features. In this manner, UAVs consume very low energy under this arrangement. Lei *et al.* in [100] modeled an energy efficient caching problem by considering data placement, user linkages, and data delivery. The authors proposed deep learning to train scheduling algorithm, by devising DNN to offer optimal and time efficient caching solutions. DNN is efficient in reducing computational load by utilizing delay sensitive applications.

Recent technological advancements in edge caching have impacts over communication in vehicular networks as well. It is highly challenging to handle evolving data popularity profile for efficient utilization of scarce resources. In addition, variable speed of vehicle increases the complexity of designing a caching mechanism. Dai *et al.* [149] proposed a novel artificial intelligence based vehicular network to achieve cooperative data caching and multipoint data transmission. An advance DRL algorithm DDPG is designed for improving resource allocation. Road Side Units (RSUs) are deployed in vehicular network for providing data services to the users along the path. However, data set size is massive and all contents cannot be cached. Therefore, the authors exploited the potential of cooperative caching in RSUs to facilitate RSUs to share data with each other to optimize data services. This enables RSUs to meet data demands of users moving with variable speeds.

2) Reinforcement Learning: Learning theoretic tools are necessary to implement proactive caching following dynamicity of data popularity profile and users' mobility patterns. Somuyiwa *et al.* in [150] proposed a proactive caching scheme for mobile users while taking ephemeral contents popularity into account. A threshold defined proactive caching scheme is designed based on MDP with side information (MDP-SI) to maximize energy conservation. The performance of proposed scheme is dependent on the CSI regarding threshold, and its value depends on the state of the system. Then, a parametric policy including Longest Lifetime In–Shortest Lifetime Out (LISO) and Linear Function Approximation (LFA) is used to achieve approximate optimality. Both of these schemes are based on low-complexity parameters of the system states and RL for optimal policy searching. Two policy gradient schemes, the Finite Difference Method (FDM), and the Likelihood-Ratio Method (LRM) are applied with non-causal knowledge of user-access times. The LFA policy provided better performance than LISO.

Networking, computing, and caching are three enabling technologies that play important role in the performance of a wireless network, but also consume a lot of energy. In [151], to reduce energy consumption, the authors put forward an energy efficient caching mechanism based on Maximum-Distance Separable (MDS) for orchestration of networking, caching, and computation. This joint optimization problem requires dynamic allocation of resources. Therefore, DRL is exploited for reducing the complexity with Deep Q-Network (DQN) and approximate Q value action function. The Google Tensorflow is used for implementing DRL.

Furthermore, Interference Alignment (IA) is also important to consider the performance of wireless networks. IA poses a significant impact on the variation in the wireless channels because network conditions varies with respect to time. He *et al.* [152] is tackled with IA issue by modeling a wireless channel as a Finite-State Markov Channel (FSMC) and a central scheduler is utilized for accumulating CSI from each user. Then, DRL using DQN is used for approximating Q value-action function. All the collected states are provided to

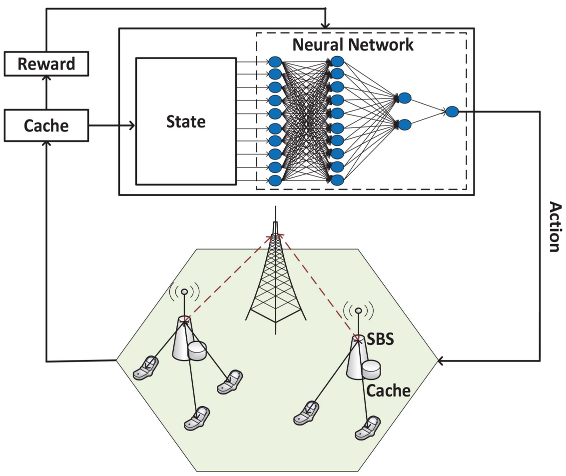

Fig. 12. An illustration of DRL based data caching in Ultra-Dense Network (UDN).

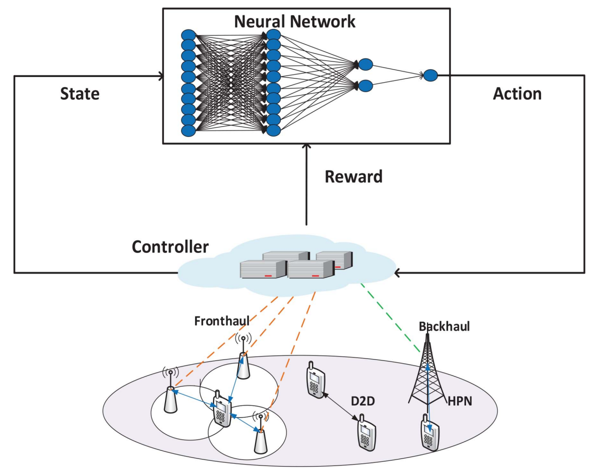

Fig. 13. System architecture of the DRL based F-RAN system.

the DQN for obtaining an optimal user selection policy and obtained an optimal interference aware user selection scheme in a cache-enabled wireless network.

Integrating caching and UDNs has the potential to provide substantial reduction in redundant data transmissions and improvement in energy efficiency. However, most of the works consider accurate prediction of data popularity profile, which is an unpractical consideration. To achieve optimal data caching, precise estimation of popularity distribution is necessary. Li *et al.* [153] proposed an online data caching policy to tackle dynamicity and lacking information of data popularity distribution. A cache-enabled UDN is considered, where multiple cache-enabled Small Access Points (SAPs) are deployed that are connected with MBS as shown in Fig. 12. The authors proposed a DRL-based caching approach that utilizes DQN for Q action-value function approximation for energy conservation based on transmission power, operational charging power, and data receiving power. An advanced DQN approach is used that consists of prioritized experience replay, dueling, and RNN. Two approaches are utilized to modify Q-learning into Deep Q Neural Network (DQNN). First approach exploits experience replay provision, while second modification devises net_target and net_evaluate NNs. The weight vector of neural network is updated for a fixed number of time steps in net_target, where net_evaluate update weight vector for each time step. This benefit of this approach lies in stability of learning procedure. The experience tuples are randomly sampled from the experience pool and actions are selected with prioritized experience replay advanced approach of DQNN for fast algorithm processing. Then, dueling network architecture is opted, which denotes Q-function as two different estimators. One of the estimators is utilized for state-value function, and the second is devised for state-dependent action advantage function. This helps to perform learning across multiple actions without any modification in the underlying RL technique. The proposed approach provided a substantial performance gain for both the stationary and dynamic data popularity distributions.

In Fog Radio Access Network (F-RAN), the communication mode and resource management becomes a challenging problem due to varying network conditions, scarce caching, and computational resources. Sun *et al.* [154] proposed a F-RAN system as shown in Fig. 13, where data caching is exploited at the users' level leveraging D2D communications yielding substantial energy conservation and alleviation of data traffic burden from the fronthaul. The authors utilized DRL approach to enable the network controller to make intelligent decisions regarding user communication modes and processors' on-off states with optimizing pre-coding for every user in order to minimize energy consumption. While energy consumption is evaluated based on total energy consumption due to front-haul data transmission, running processors, and users' wireless communications. DRL approach trains the network controller through raw data to opt intelligent decisions yielding energy conservation. The problem is formulated through MDP, where system state consists of processors on-off states, current data transmission modes, and cache state of all data transmitting users. This information is provided to the DRL to generate output values corresponding to the action. ϵ greedy policy is devised to perform an action in terms of either to turn on or off processor, change in data transmission mode. All the interactions regarding state transitions, action, and energy consumption (reward) are saved in controller's replay memory. After several interactions, training is performed over a batch of interactions, which are randomly sampled from the replay memory in order to minimize error between target Q-value and the predicted Q-value.

D. AI-Based Caching for Optimizing Delay

A user data service experience is mainly reposed on the network's jitter or delay that evaluates Quality of Service. Therefore, to optimize network delay several AI-based caching approaches are exploited to improve data caching by observing users data request patterns, mobility patterns, and evolving data popularity profile. In the following, we provide detail discussion of several AI-based caching approaches that target minimization of network delay.

1) Supervised Learning: Caching in D2D networks is a prodigious way of leveraging data sharing among users.

However, users are selfish and they are not willing to participate in improving network performance at the cost of their own scarce resources such as battery and bandwidth. The D2D communications have a tremendous potential to maximize data access, therefore, schemes are needed to motivate users to play their role in data caching and sharing. Ma *et al.* [155] exploited distributed caching mechanism for improving D2D communications. The characteristics of social networking are exploited for dealing with selfish nature of users that do not like to take part in caching process at the cost of their memory and energy resources. For making adaptive decisions utilizing the RL approach, a discrete generalized pursuit algorithm with social characters (DGPA-SC) is proposed. This algorithm optimizes content placement while satisfying social and physical constraints. The learning automaton facilitates users to cache contents for improving rate of caching necessary contents. The users or communities are partitioned based on the similarities in their locations and interests. Then, cluster centroids or influential users are selected that cache contents and share contents with other users through D2D communications. The proposed scheme results depict low transmission delay and high convergence speed.

Traditional data caching approaches require accurate prediction of data popularity profile and dynamics of cellular networks. The content popularity is determined based on the data requests statistics of each content. However, it becomes a challenge to predict content popularity without knowing data request statistics. Tanzil *et al.* [156] studied the access behavior and characteristics of cellular networks (i.e., cache capacity, data traffic load, and bandwidth). Therefore, the authors proposed Extreme Learning Machine (ELM) feed-forward NN and devised perturbation stochastic approximation to minimize the neurons quantity without effecting prediction of data popularity profile accuracy. Since estimation of content popularity is performed during minimal data traffic, it enabled efficient usage of network resources during peak traffic hours.

2) Deep Neural Network: Tsai *et al.* [157] proposed context-aware data caching in the Heterogeneous Small Cell Networks (HSCNs) for reducing the service delay for the end users. The data request prediction is performed to reduce service delay by placing predicted contents close to the end users. The data caching is proposed in three cache entities including edge caching elements, SBSs, and Macro Base Stations (MBS). Then, CNN is utilized for data extraction and analysis. First, data preference list is computed for predicting future data requests. Then, all the contents are ranked by the CNN classifier. Next, an average value of the products of the classification and confident level for all the classifiers is calculated for obtaining final data classification. Subsequently, a College Admission (CA) based data caching model is used for proposing data to most preferred cache entities. Finally, cache entities determine whether to accept the data or not based on its preference and capacity limitation. Hence, this approach facilitates an optimal prediction of data requests and service delay can be minimized by placing contents close to the users.

The prediction of user playback behavior (time spent on any particular content) is complicated especially in case of video streaming. Deep learning methods unlike traditional methods that uses known features, trains a multi-layer neural network to optimize a model based on loss function. It appears to be a suitable solution to deal with dynamic and complex data caching problems. Hao *et al.* [158] utilized knowledge centric edge data caching. First, user semantic information is captured by DBN. DBN is a graphic model that belongs to a class of deep neural network. It is used to determine the output of a probability model in order to know latent features. The performance of DBN can be improved by incorporating back propagation of the regrets between actual and predicated outcomes. Then, data caching optimization problem is formulated for future requests prediction. Finally, a greedy algorithm is employed for reducing data delivery delay in case of video streaming.

A learning based proactive caching scheme devise Singular Value Decomposition (SVD), which unifies missing values that leads to uncertain predictions. In addition, SVD can observe the approximated series of elements incurring negative numbers causing a lack in information regarding users' requests. Deep Learning (DL) relying on DNN is a promising solution that learns by devising multiple data processing layers. Saputra *et al.* [159] devised DL to determine users' data requests trends to optimize proactive data caching. DL is applied on the collected information in a centralized manner. Then, Distributed Deep Learning (DLL) is exploited at the edge for sharing information for minimizing errors in the prediction of data request patterns. The proposed edge caching scheme exchange users' gradient information while maintaining privacy of mobile users.

3) Reinforcement Learning: For reducing delay of the cache-enabled wireless networks, one of the enabling mechanisms is to place data close to users in order to avoid data fetching from the core network. The geographical impact on the popularity of contents is necessary to monitor because users belonging to specific regions may have relations in their data preferences. Although users are mobile and they have frequent roaming across different cells. It is essential to design such a system that can analyze users data preference on the fly. Jiang *et al.* [160] utilized Multi-agent reinforcement learning approach to cache contents in UEs without knowing data popularity profile for minimizing network delay. Users play the role of agents that learn Q-values in feedback of its own actions along with actions of other agents. Since the action space is exponential to the number of agents and contents that creates a huge action space, however, traditional multi-agent Q learning algorithm cannot deal with it. Therefore, a Belief-based MCUCB algorithm is devised to reduce action space. Furthermore, belief-maintenance is used to deal with the policies of other agents based on their beliefs of policies of other agents. For optimizing D2D communications, it is essential that users act as ILs to learn Q-values of their own actions in conjunction of other users as well. However, the caching decisions are only justifiable when no user changes its caching decision, where adaptive caching decisions may lead to errors in the caching decisions. Jiang *et al.* [161] proposed Joint Action Learners MCUCB (JAL MCUCB) algorithm that helps each user to determine Q-values of their own actions in conjunction of other users' actions. This approach significantly reduces computational complexity caused due to large size of action space.

The data similarities among users following sociality, data interests, and grouping can also be used to optimize network delay. The users' grouping phenomena is used in [106], where authors performed content-aware user clustering and data caching at the SBS level. The problem is divided into two sections. First, based on the similarity in content interests, users are grouped via spectral clustering algorithm for associating similar content interest users with the same SBS. Second, RL is proposed for enabling SBSs to learn content popularity by monitoring the content request pattern of the users associated with it. The SBSs learn probability distribution of cached contents by minimizing regret over caching contents in the past and utilizing the knowledge for improving caching decisions in future time instants. For dynamic update of caching policy, a regret minimizing learning algorithm is proposed. The results show that a higher offloading gain is achieved and thus the delay is minimized, by correlating different popularity patterns of individual users.

It is imperative to design caching mechanism based on content dynamics, spatio-temporal data demands, and data service capacity constraint. All the mentioned information is unavailable. However, machine learning approach can be exploited to learn network dynamics based on historic information to increase the probability of cache hits. Jiang *et al.* [162] proposed a MARL based data caching approach that does not require prior information regarding data request patterns and caching decisions are performed based on historic data demands. To achieve coordination in caching decisions in a multi-agent environment Q-learning is utilized. The complexity of traversing large number of Q-values is reduced by utilizing confidence bound approach that significantly minimizes Q-table space.

There is a tremendous development in cloud-based IoT. Despite of its utility in providing better QoS to users, it contains several challenges such as latency and backhaul bandwidth. Thus, orchestration of computation, caching, and communications is necessary to improve network performance in IoT. Wei *et al.* [163] focused on optimal data caching and user scheduling. Initially, information regarding network states is unknown, therefore, RL is employed to learn favorable stochastic policy incurred due to interaction between agent and environment. The actor uses Gibbs distribution to perform probabilistic caching and policy is updated via Gradient ascent technique. The DNN is devised for accurate estimating of the value functions in the critic portion because of a huge size of state and action space. Moreover, DNN is utilized for parameters stochastic policy, which is improved by critic. The convergence of the mechanism is ensured by devising target network and experience replay.

4) Transfer Learning: Bharath *et al.* in [164] considered HetNets containing MBSs, SBSs, and users are distributed according to independent PPP. The popularity profile of data is estimated through instantaneous content requests of the users. The learning is performed during the training phase for achieving offloading loss $\epsilon > 0$ near to the optimal policy. If the number of users is above than threshold, then the training time is finite for achieving $\epsilon > 0$ difference between the achieved cost and optimal cost. TL is used for improving the accuracy of estimation and reducing training phase by modeling data popularity profile through parametric series of distribution. The delay scales linearly with distribution parameter dimensionality.

E. AI-Based Caching for Optimizing Throughput

The performance of a wireless network is heavily affected by the amount of data that can be successfully transmitted from source to content demanding user within a certain period of time. We discuss below, AI-based caching approaches focusing on maximizing network throughput.

1) Supervised Learning: The global popularity of contents does not take individual users' content preferences into account, which causes reduction in the efficacy of the caching mechanisms. For better utilization of limited cache resources Cheng *et al.* [165] utilized Bayesian theory to predict users' data preferences. Individual Content Request Probability (ICRP) is estimated for improving network throughput. This method is known as Constrained Bayesian Probabilistic Matrix Factorization (CBPMF). This method takes rating matrix imbalance into account for prediction of accurate ratings. This helps to translate knowledge of global and local ratings for estimating ICRP. Then, reinforcement learning based Deterministic Caching Algorithm (DCA) is used for improving data placement in UEs. Moreover, the proposed algorithm enables adjusting probability of each content caching.

2) Reinforcement Learning: Data caching is an economical way to provide data services to alleviate network traffic congestion in throughput constrained backhaul links. Meeting more data requests via scarce cache resources, the network throughput is benefited considerably. Since all the contents cannot be placed in the limited cache resources, an appropriate data placement is crucial for improving network throughput. Gao *et al.* in [166] investigated resource allocation problem for cloud based cache-enabled small cell networks. The authors proposed data caching both at the cloud pool and each SBS. LSTM model is used to predict users' mobility pattern and evaluate allocation resources. First, the NN framework of LSTM performs prediction of the users' mobility pattern and determine the users' association. Then, a RL-based resource allocation algorithm is devised for maximizing the network throughput. The results show high throughput gain compared to random and the nearest algorithms.

He *et al.* in [167] considered time varying channel of a cache-enabled wireless network. FSMC is used for modeling the channel dynamics. But the complexity of FSMC is very high, which is encountered by utilizing DRL. The benefit of DRL is in approximating Q value-action functions by using DQN. A central scheduler is used for collecting all CSIs of a time varying channel. All gathered information is forwarded to the DQN to evaluate optimal user selection policy. This improves interference alignment, user selection, and network throughput.

F. AI-Based Caching for Optimizing Cost

In this subsection, we discuss AI-based caching techniques to minimize cost (i.e., bandwidth consumption and signaling overhead etc.) of wireless networks .

1) Deep Neural Network: The content popularity profile has a great impact on the spatio-temporal distribution of content placement and data transmission relies on its effectiveness. Liu *et al.* in [168] proposed a Deep-Learning Content Popularity Prediction (DLCPP) based on Software Defined Network (SDN) for data popularity profile prediction. This scheme performs switching of links and computing resources for building a distributed and reconfigurable deep learning network. First, the variation in data popularity profile is observed. Then, each node gathers information regarding spatial-temporal distribution of data popularity. All the information is fed as an input to Stacked Auto-Encoders (SAE) for extraction of spatio-temporal characteristics of data popularity. At last, the prediction of content popularity is transformed into a problem of multi-classification by performing discretization of content popularity into multiple categorizations for numerical evaluation.

2) Reinforcement Learning: Wang *et al.* in [169] focused on minimizing the transmission cost by data caching at BSs. The caching problem is modeled as MDP, then Q-learning based distributed cache replacement mechanism is proposed. Each BS senses its current state and based on the current state performs an action based on the current state. Each action infers transition of network to a new state. As a result of each action, a reward is given to the BS. An optimal policy is defined for achieving a high reward without prior knowledge of the system. The Q-value is estimated for all the state-action pairs based on the current state of the BS, future state in result of transition and the probability of transition between states. The authors observed cellular serving ratio under static and dynamic network conditions in order to measure data offloading via cache resources. The proposed approach outperformed state-of-the-art schemes including random, LFU, and LRU caching. Since SBSs can be equipped with limited cache resources, cache refreshing is essential.

In [55], transmission cost minimization is performed by optimizing the content replacement in SBSs. The authors modeled content replacement problem as MDP, which enables BSs to sense their present state and select an action from the action set. In result, the environment makes transition to a new state and get a reward as a feedback. For defining an optimal policy set Q-learning is utilized for achieving accurate distributed content replacement in order to minimize transmission cost of wireless networks.

Global popularity of data cannot optimally determine popularity of data due to dynamic geographical and temporal features. In [170], user requests are modeled by (i) a local and global Markov processes model, and (ii) RL with Q-learning algorithm. The Q-learning algorithm is used to determine the optimal caching policy in an online manner, and thus offers an asynchronous caching. This enabled SBSs to learn and track variations in the popularity of contents. For achieving fast convergence and reducing complexity Q-learning approach is opted and spatio-temporal popularity of data requests is considered. This caching mechanism uses cache in SBSs instead of core networks, in order to meet users' data demands.

Most of the researchers considered deterministic parameters to simplify the problem, however, such consideration is inaccurate to design practical systems due to the system failure risk. The practical systems rely on time-varying spatio-temporal data popularity, physical layer conditions (e.g., bandwidth, channel capacity, traffic and channel fluctuations). Sadeghi *et al.* [171] necessitates stochastic optimization perspective while targeting minimization of the aggregated cost based on performing flexible data fetching and caching with respect to time. The objective is to perform real time data fetching and caching decisions in order to minimize both the expected present and future costs. Therefore, caching decisions are performed based on every slot time, which influences future cache states. First, costs and data popularity profile are considered to follow stationary distribution. Then, cost and cache optimization problem is solved by employing an online low complexity Q-learning based RL algorithm for optimal data fetching and caching decisions, which provides convergence of stochastic estimates. To overcome the curse of large dimensions, the authors defined marginalized Q-function that depends on the state and immediate actions. The advantages of exploiting stochastic methods are that it does not require complete knowledge of state distribution, it can easily handle dynamic environments, and maintains complexity to a low level.

Moreover, Tan and Hu [172] targeted cooperative caching and computation in vehicular networks taking vehicles mobility and delay constraints into account. Therefore, an orchestration of caching, computation, and mobility is exploited to reduce cost. The authors performed cooperative coded data placement and allocation of computation resources at vehicles and RSUs by incorporating mobility feature for designing resource allocation policy that posed severe complexity. Vehicles can access data from vicinity vehicles or RSU. In addition, vehicles can also offload their tasks to other vehicles or RSUs. This gives rise to action space set. Therefore, deep Q learning is used, which stores agent experiences in the replay memory and Q-network is updated at each time step with sample batches from the replay memory. They also utilized the ϵ-greedy scheme to achieve a balance between exploration and exploitation, and thus maximized rewards of computation and data access services.

For maximizing network performance and minimizing cost, it is essential to lower the uncertainty of data demands by maintaining a tradeoff between using known policy that yields a large profit and searching new policies that can provide more higher profit to the network operators. Alotaibi *et al.* [173] focused on meeting users' data demands for maximizing network's profit. Since it is necessary to provide incentives to the users taking part in proactive caching, smart pricing is performed by making a tradeoff between exploration and exploitation. A RL approach is used to reduce the regret by announcing a price-pair for improving cumulative expected

TABLE IV
ARTIFICIAL INTELLIGENCE (AI) BASED DATA CACHING FOR PERFORMANCE OPTIMIZATION OF WIRELESS NETWORKS

Ref.	Year	Technique	Performance Metric	Caching Level	Problem
[124]	2017	PAC	Offloading	SBSs	A high probability bound on data traffic offloading loss is evaluated by PAC in a situation of varying data popularity.
[125]	2019	LSM	Offloading	UAV	LSM is devised to jointly optimize data caching and resource allocation.
[126]	2017	k-NN	Offloading	SBSs	k-NN is utilized for observing spatio-temporal behavior of data demands.
[127]	2019	DL	Offloading	SBS	DL is utilized for extraction of hidden features of data and users.
[128]	2016	ESNs	Offloading	RRHs	ESNs is used for predicting distribution of content requests with limited network and user states.
[129]	2017	ESNs	Offloading	RRHs	ESNs is utilized for predicting users data request distribution and mobility pattern with limited information about network and user states.
[130]	2014	MAB	Offloading	SBSs	Optimal data caching is performed by utilizing MAB approach without prior knowledge about contents popularity.
[131]	2014	MAB	Offloading	Central controller	MAB is used for refreshing cache and perform cache placement based on previous history of users data access behavior.
[132]	2018	DRL	Offloading	BS and UEs	Exploited social relationships among users and Deep reinforcement learning is employed for automatic decision making for resource allocation.
[133]	2015	TL	Offloading	SBSs	TL is utilized for improving accuracy of the estimation of data popularity profile.
[134]	2015	TL	Offloading	SBSs	TL is used for exploiting contextual information of users such as social relationships and data history etc.
[135]	2017	Supervised learning	Cache hit rate	Edge	A self-starting offline user preference model based on supervised learning approach is used for updating contents by observing logistic loss.
[136]	2018	ESNs	Cache hit rate	SBSs	ESNs predict data popularity and based on prediction most popular contents are cached.
[137]	2019	Bidirectional DRNN	Cache hit rate	BS	Bidirectional DRNN model is utilized to improve the prediction of time-series data requests and data caching.
[138]	2018	RL	Cache hits	Node	RL is utilized for data caching in an online manner.
[139]	2016	Q-learning	Cache hit rate	Source server	Q-learning is used for data replacement in a distributed cache environment.
[140]	2017	Q-learning	Cache hit	Virtual machine	Hierarchical structure of RL is used for abstracting features.
[141]	2019	Q-learning	Cache hit rate	UEs	DRL is devised to optimize power allocation and handle noise in the training data set.
[142]	2016	MAB	Cache hit	SBS	MAB approach is devised to update cached contents and data requests patterns based on contextual data popularity profile.
[143]	2017	MAB	Cache hits	SBSs	MAB is devised for determining varying popularity of contents while taking intermittent connectivity among users due to mobility.
[144]	2014	MAB	Cache hit rate	SBS	Data coding is performed using Fountain encoding and MAB approach is used for data placement in SBSs.
[145]	2018	DRL	Cache hit rate	BS	DRL is used for defining the state and action spaces, and reward for the agent to achieve optimal caching.
[146]	2014	DRL	Cache hit rate	BS	DRL is used for designing an optimal data caching based IoT framework without prior knowledge of the IoT system .
[147]	2019	DQL	Cache hit rate	SBS	Network performance is improved based on Frequency Maximum Q-value (FMQ) and distributed Q-learning technique is used to optimize data caching
[148]	2019	DQL	Cache hit rate	SBS	DQL approach is used to optimize data caching while considering dynamic user data preferences and data popularity profile.
[149]	2017	ESNs	Energy	Unmanned aerial vehicles (UAVs)	ESN based learning algorithm predict distribution of data request of mobile users following their context.
[101]	2017	DNNs	Energy	SBS	DNN trains scheduling algorithm for optimal and time efficient cache decisions.
[150]	2019	DDPG	Energy	BS and RSU	DDPG approach is used to perform orchestration of computing and caching resources to optimize data offloading, data caching, and delivery.
[151]	2018	RL	Energy	UEs	RL is utilized for optimizing thresholds for pushing contents towards cache depending on channel status with respect to the threshold value.
[152]	2017	DRL	Energy	BS	DRL is used for resource allocation for performing orchestration of communication, computation, and caching in HetNets.

profit. The prediction of content requests is harnessed for performing proactive data caching in order to minimize expected payments. An iterative gradient based scheme is utilized for reducing the regret because both regret and cumulative expected profit are inversely proportional to each other.

Proactive data caching has the potential to provide better data service under poor channel conditions. However, fluctuation in channel conditions makes caching a challenging task. To deal with this problem, Guo *et al.* [174] proposed DRL for dynamic resource allocation without channel and achievable data transmission rate information. MDP is used to jointly model bandwidth allocation and cache management to maximize data streaming for every user. Then, the authors proposed DRL for maximizing reward when data streaming exhibits neither underflow nor overflow. First, cache-aware data streaming is designed by exploiting available cache spaces at each UE. The reward function is defined as efficient data streaming when neither playback buffer overflow nor underflow occurs. DRL is utilized to optimize bandwidth and cache resources based on previous experience without any prior information regarding CSI, cache state, and data transmission rate.

To meet an unprecedented growth in IoT users demands, many efforts are performed by the research community to perform resource allocation to achieve a high QoE. An innovative approach is devising green QoE driven solutions in IoTs. He *et al.* [175] discussed data placement in the cache resources and data transmission rate to optimize QoE. The QoE is measured through users' experience and network cost. First, dynamic data transmission rate is devised to improve QoE by utilizing Shortest Path Tree (SPT) algorithm. Then, node centrality is exploited for improving content delivery in the wireless networks by optimizing resource allocation through SPT. Moreover, DRL is proposed to achieve green resource allocation for improving QoE in an adaptive manner. The system state is modeled through MDP, where the state constitutes of cache nodes and content transmission rate. In each state, an action is performed regarding achievable transmission rate and cache decisions. Then, DQN is devised to perform intelligent decisions regarding resource allocation in an online random environment.

3) Transfer Learning: The diversity and dynamicity of users' data demands require monitoring of spatio-temporal features of each user. To achieve this monitoring, the system needs to be trained with high dimensional data to accurately estimate data popularity profile based on users' preferences. TL can provide an alternative as it can utilize training of another system to train a new but similar system. Hou *et al.* in [176] overcame the challenges of developing cooperation among distributed cache resources by adopting proactive caching. The authors proposed a Learning-based Cooperative Caching (LECC) for minimizing content transmission cost by increasing user QoE. The problem of optimal content caching is divided into two parts. First, TL is used to predict the content popularity. Then, a greedy algorithm is proposed for achieving content placement.

Summary: This section elaborates the implementation of AI approaches for data caching in wireless networks. The performance of AI-based data caching schemes depends on data popularity profile, association between users and BSs, and social relations among users. The discussed works in this section cover a wide range of AI-based caching techniques to optimize key performance indicators (e.g., offloading, cache hits, delay, throughput, energy, and cost). The contributions are summarized in Tables IV and V, presenting state-of-the-art AI algorithms for intelligent data caching. The supervised learning based caching schemes require high computation complexity during training procedure. For instance, NNs can perform BSs clustering by exploiting data request distribution and mobility patterns. The need of prior knowledge is eliminated with the utilization of unsupervised learning, RL, and TL approaches for data placement. RL performs data placement without knowing users data interests. On the other hand, RL can also be combined with NNs, which is known as DRL to improve predictions accuracy. Whereas TL may learn users' contextual information in one environment and can utilize this learning to train other similar environment. This eliminates the need of new dataset for training every environment, which reduce the complexity and optimizes QoE by controlling the reward functions.

V. Summary of Lessons Learned, Challenges and Future Research Directions

In this section, first we discuss the lessons learned. Then, we highlight existing challenges and future research directions.

A. Lessons Learned

The classification and regression evaluation are the two main approaches of supervised learning that can be utilized for determining data popularity profile and user preferences of mobile users. A dataset is required in supervised learning to take better data caching decisions. However, it is a cumbersome task to handle large dataset or training the system with insufficient information. This problem can be resolved by utilizing unsupervised learning. Unsupervised learning enables clustering of dataset to highlight multiple patterns. Clustering process can benefit data caching by partitioning users in groups based on the data request history and social relations among users [106]. On the other hand, RL enables an agent to take decisions autonomously following obtained rewards in result of actions. For instance, Q-learning can be devised to predict data popularity distribution, data arrival, user arrival, etc. by finding optimal Q-values. TL utilizes known information for solving one problem to solve other similar problem. TL may help to perform data caching in a wireless network by using the knowledge gained from other network such as social network. This improves the estimation of data popularity and caching mechanism [164]. The shortcoming of TL approach is its requirement of a large number of relations between target and source domains. This dependence on a large number of relations between two domains to satisfy correlation, degrades the performance of data caching mechanism. Most of the researchers have devised Deep Learning for data caching due to its hierarchical architecture. This yields precise cache decisions following feature extraction from the dataset with significant reduction in complexity.

B. Challenges and Future Research Directions

Despite several efforts in optimizing network performance many challenges still need to be addressed. In this section, we discuss the challenges and future research directions.

1) Data Integrity: Wireless networks contain a large number of users and each user needs to be authenticated for avoiding threat to the users' profile that is constitute of private and confidential information. While the existing privacy ensuring mechanisms do not consider intermittent connectivity of mobile users, which gives raise to the challenge that how the data caching and transmission can be done without information compromising. When we apply any learning tool for training purpose, the outcome of the training data may contain private and volatile personal information. The leakage of such private information may cause security and privacy threats to the users. For instance, in training phase a system gathers

TABLE V
ARTIFICIAL INTELLIGENCE (AI) BASED DATA CACHING FOR PERFORMANCE OPTIMIZATION OF WIRELESS NETWORKS

Ref.	Year	Technique	Performance Metric	Caching Level	Problem
[153]	2017	DRL	Energy	BS	Channel state information for each user in the network is calculated and all the information is provided to the DQN for optimal user selection to perform data caching.
[154]	2019	DRL	Energy	SBS	DRL based on DQN is used to improve data caching decisions based on information stored in experience replay
[155]	2018	DRL	Energy	UE	DRL model is trained for intelligent decision making and TL is incorporated for fast processing.
[156]	2016	LA	Delay	UEs	Learning automaton is utilized to determine social characters to resolve selfish behavior of users and help users to store data.
[157]	2017	ELM	Delay	BS	ELM is devised to predict data popularity profile while exploiting human perception models
[158]	2018	CNN	Delay	Edge	CNN model is used to extract features of data to predict data requests made by the users.
[159]	2018	DNN	Delay	BSs	Deep Belief Networks (DBNs) a class of DNNs is used for capturing users' semantic information of playback behavior.
[160]	2019	DL	Delay	SBS	DL approach is exploited to share information for minimizing error in data demands prediction while maintaining privacy
[161]	2018	Q-learning	Delay	UEs	Q-learning is used for coordination of caching decisions in a multi-agent environment.
[162]	2019	RL	Delay	UEs	JLA MCUCB algorithm is proposed by determining Q-values of each user action in conjunction of other users' actions .
[107]	2014	RL	Delay	BS	RL is used to determine distribution of content requests by cluster of users.
[163]	2019	Multi-agent MAB	Delay	MEC server	Multi-agent MAB approach is used to perform cooperative data caching
[164]	2018	Actor-critic DRL	Delay	SBS	Reinforcement learning is used to jointly address data caching and user scheduling problems.
[165]	2016	TL	Delay	BS	Transfer learning is employed to achieve finite time training as a function of distance between source domain samples and probability distribution of requested contents.
[166]	2018	Bayesian learning	Throughput	BS	Bayesian leaning is used to predict user individual data preferences and estimate ICRP by observing imbalance in the rating matrix. This enables accurate prediction of unknown ratings
[167]	2017	RL	Throughput	BS	RL is used for autonomous decision making for allocating resources based on network status
[168]	2017	DRL	Throughput	BS	DRL provides optimal interference alignment user selection scheme by using DQN that approximates Q value-action function
[169]	2018	Deep learning	Cost	Nodes	Deep learning couples network and application characteristic and unite data collection and analysis.
[170]	2017	Q-learning	Cost	BSs	A distributed caching is performed based on Q-learning for replacing old contents with new and more popular contents
[56]	2014	Q-learning	Cost	BSs	A distributed caching mechanism is used based on Q-learning approach following data popularity and cost of transmission between two BSs
[171]	2018	Q-learning	Cost	BS	Q-learning is used to learn policy for entailing estimation of data popularity profile when underlying states are unknown
[172]	2019	Q-learning	Cost	SBS	Q-learning is devised to optimize data fetching decisions
[173]	2018	DQN	Cost	Vehicles and RSUs	Q-learning framework is used for configuring communication, data caching and computation resources in vehicular networks
[174]	2016	RL	Cost	UEs	Reinforcement learning is used to investigate how dynamic pricing can maximize profit without prior knowledge of user demands
[175]	2019	DRL	Cost	BS	DNN to optimize storage management and bandwidth allocation policy
[176]	2018	DRL	Cost	SBS	DRL approach is used to improve resource allocation to optimize QoE.
[177]	2018	TL	Cost	Server	Transfer learning is used in LECC for data popularity estimation

information regarding users' position, data viewing patterns, social and proximity relationships, etc. In case of lacking data protection mechanisms, users may experience serious problems. Thus, implementation of security mechanism is imperative for learning based caching mechanisms to develop a system considering authentication, data integrity, and users' confidentiality, which enables secure data caching and data accessing.

2) Dynamic Environment: Mobility is neglected in most of learning based caching approaches. The network performance can be optimized by taking users' mobility patterns into account. This enables caching at BSs following data demands of existing users under its coverage. Exploiting mobility in D2D communications is more necessary because of intermittent connectivity between mobile users. Caching strategies in UEs can also take advantages of social and geographic information (e.g., route, weather, traffic congestion). However, when users exhibit high mobility, they may experience severe dynamics and connectivity issues caused by rapid variations between the pair of users. Therefore, meticulous efforts are required to observe users' mobility patterns. In such situation, AI can be utilized to estimate each user next location by learning users' mobility patterns. The network resources can be reserved by learning the stochastic behavior of users' arrivals and departure. Hence, AI can reduce signaling overhead and substantially improve the network utility.

3) Utilization of Big Data: In wireless networks, huge data is generated with every passing time. The data comprises of content feature, popularity, user preferences, transmission parameters, and user locations. Therefore, network operators are considerate towards big data analytics to perform better network planning for efficient utilization of network resource and reduce the operational cost. This requires intelligent big data analytics to improve caching mechanisms in wireless networks. However, AI-based big data analytics possess challenges of handling large datasets and at the same time expensive computations are required for feature extraction of input datasets. Also, the network computing resources are not sufficient for processing high-dimensional dataset. This raises a serious question of whether all the information needs to be processed or only samples from a large dataset can be utilized for the training purpose. AI is promising in minimizing the computational cost of training phase while dealing with high-dimensional input dataset.

4) Learning-Based Caching Strategies: There are several challenges when considering data caching at UEs such as determining popularity of contents, identifying users' data preferences, and social relationships among users. Learning mechanisms require collection of all this necessary information regarding training. For an efficient caching at the UE level, it is essential that a user has access to the status of all its neighboring users within the D2D communication range. However, users are mobile and connectivity among users keeps varying due to which it is difficult to acquire all the required information. In addition, UEs have limited processing capabilities and exhibit mobility that limits their learning process. The potentials of AI should be leveraged to perform learning-based caching strategies to exploit social and spatial structure of wireless networks. Furthermore, AI approaches can be used to bound computational complexity and provide accurate results based on large multi-dimensional contextual information (i.e., users' social relationships and data viewing preferences).

5) Optimizing Mobile Crowdsensing (MCS): It is a promising model for enabling massive sensing of environment. It can extend sensing by jointly utilizing offline (i.e., UEs sensed information) and online (i.e., social network) information. With the ongoing technology advancement, the existing systems have the potential to improve data traffic planning, data planning, and recommendation based on sociality information. MCS allows aggregation of a large amount of information by the users regarding local information, user data demands, and traffic conditions. Caching can be performed in the MCS environment to store huge amount of sensed data. There are two types of data to be cached. First type of data needs to be utilized immediately for current tasks and the second type of data is for future usage. However, existing caching mechanisms perform data caching that can be used for the current or future utilization, which may cause wastage of scarce cache resources. Therefore, AI can be leveraged for intelligent caching for accurate decisions regarding which data should be cached or discarded in order to save energy and limited cache resources.

6) Management of Physical Layer: In 5G, application of multiple waveforms is envisaged based on user and application requirements and network channel conditions. This requires learning of network dynamics and auto-selection of the waveforms for a particular network environment. For instance, when physical layer features such as multipath fading is taken into account, then caching same content at different transmitters can facilitate channel diversity gain. However, conventional caching approaches do not consider physical layer characteristics of wireless networks to maximize the total number of stable queue users, i.e., the performance metric to evaluate the dynamic data transmission rate of every user. Hence, to ensure a high network performance, a robust mechanism is needed for autonomous selection of modulation and coding schemes to perform data caching in such a manner that can improve the data transmission at any given time. AI can be utilized to learn users' behavior, network dynamics, and channel conditions such as interference and coding scheme in order to optimize physical-layer aware data caching in wireless networks.

7) Caching for mmWave Networks: There is a major portion of underutilized spectrum, substantial efforts are carried out to utilize the benefits of mmWave in order to optimize spectrum usage. Despite the mmWave's large bandwidth, its networks potential is restricted due to users' mobility and high path loss. Moreover, the mmWave band communication is highly sensitive to physical environment conditions, such as mmWave signals are heavily attenuated by concrete buildings and road infrastructures. Hence, its optimal performance is only confined to line-of-sight communication links. Therefore, duplication of content placement at various cache resources is necessary to reduce the probability of cache hit misses. Since content placement is based on the prediction of data popularity, it is important to explore accurate prediction model using various information, e.g., user social information, data request, and user mobility. Then, based on the prediction model, AI approach can be devised as a tool to achieve a balance between network resources and its performance.

8) Data offloading via Caching: Data caching at the network edge can offload data traffic through the cache resources, which can reduce network traffic overhead during peak hours. Therefore, network operators need to jointly take data caching and routing policy into account to optimize data offloading through the cache resources in order to alleviate network congestion [177]. Existing caching mechanisms cannot perform accurate prediction of users' data viewing patterns while considering personalized QoE. Since users' data preferences and data service requirements are heterogeneous, caching policy should be defined under realistic network features to improve the caching performance. AI can accurately determine contextual information such as users' data viewing patterns, social ties, data sparsity, and spatio-temporal data demands. Then, based on the extracted data features optimal data caching can be performed, which will provide substantial improvement in the data offloading through the cache resources.

VI. Conclusion

AI-based data caching is an autonomous mechanism to improve accuracy of estimation of data demands and popularity of contents that is helpful in effective data caching for increasing users' QoE. We discussed fundamental modes of data caching, i.e., centralized and distributed data caching. In addition, we highlight the performance metrics and limitations in the traditional caching approaches. Then, we provided an overview of several state-of-the-art AI algorithms encompassing supervised learning, unsupervised learning, reinforcement learning, and transfer learning. After providing fundamental concepts of AI approaches, we categorized and introduced significant works utilizing AI approaches for data caching in wireless networks. Finally, we discussed the challenges and future research directions. AI has shown significant improvements in the networks performance. The preliminary steps are taken in the realm of AI-based data caching and no practical or industrial implementation is documented yet. For future wireless networks, it is necessary to design an AI-based caching mechanism by considering data secrecy, mobility, content popularity, user demands, and limited cache resources. This requires collection of large amount of information regarding user preferences, mobility patterns, network conditions, and evolving data. AI can play a significant role in extraction of useful information and provide efficient services. Hence, development in this area requires strong tie between academia and industry. In this paper, we attempted to provide an insight to the readers in order to provide key concepts of data caching, AI approaches, and AI implementations in cache-enabled wireless networks.

REFERENCES

[1] S. Barbarossa, S. Sardellitti, and P. Di 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.

[2] *Global Mobile Data Traffic Forecast Update, 2015–2020 White Paper*, document 958959758, Cisco, San Jose, CA, USA, 2016.

[3] M. Peng, Y. Li, J. Jiang, J. Li, and C. Wang, "Heterogeneous cloud radio access networks: A new perspective for enhancing spectral and energy efficiencies," *IEEE Wireless Commun.*, vol. 21, no. 6, pp. 126–135, Dec. 2014.

[4] L. Chen, L. Song, J. Chakareski, and J. Xu, "Collaborative content placement among wireless edge caching stations with time-to-live cache," *IEEE Trans. Multimedia*, vol. 22, no. 2, pp. 432–444, Feb. 2020.

[5] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1473–1499, 3rd Quart., 2015.

[6] J. Kwak, Y. Kim, L. B. Le, and S. Chong, "Hybrid content caching in 5G wireless networks: Cloud versus edge caching," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3030–3045, May 2018.

[7] M. Ahmed, Y. Li, M. Waqas, M. Sheraz, D. Jin, and Z. Han, "A survey on socially aware device-to-device communications," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2169–2197, 3rd Quart., 2018.

[8] B. Chen and C. Yang, "Caching policy for cache-enabled D2D communications by learning user preference," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6586–6601, Dec. 2018.

[9] Y.-J. Yu, T.-Y. Hsieh, and A.-C. Pang, "Millimeter-wave backhaul traffic minimization for CoMP over 5G cellular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4003–4015, Apr. 2019.

[10] S. Biswas, T. Zhang, K. Singh, S. Vuppala, and T. Ratnarajah, "An analysis on caching placement for millimeter—Micro-wave hybrid networks," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1645–1662, Feb. 2019.

[11] N. Deng and M. Haenggi, "The benefits of hybrid caching in Gauss–Poisson D2D networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1217–1230, Jun. 2018.

[12] X. Zhang and Q. Zhu, "Collaborative hierarchical caching over 5G edge computing mobile wireless networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–6.

[13] B. Tang, H. Gupta, and S. R. Das, "Benefit-based data caching in ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 3, pp. 289–304, Mar. 2008.

[14] C. Yi, S. Huang, and J. Cai, "An incentive mechanism integrating joint power, channel and link management for social-aware D2D content sharing and proactive caching," *IEEE Trans. Mobile Comput.*, vol. 17, no. 4, pp. 789–802, Apr. 2017.

[15] B. Bai, L. Wang, Z. Han, W. Chen, and T. Svensson, "Caching based socially-aware D2D communications in wireless content delivery networks: A hypergraph framework," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 74–81, Aug. 2016.

[16] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.

[17] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[18] S. Karunaratne, R. Atawia, E. Perenda, and H. Gacanin, "Artificial intelligence driven optimization of channel and location in wireless networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–6.

[19] 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.

[20] E. Ko and K.-C. Chen, "Wireless communications meets artificial intelligence: An illustration by autonomous vehicles on Manhattan streets," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–7.

[21] S. D. Milner, C. C. Davis, H. Zhang, and J. Llorca, "Nature-inspired self-organization, control, and optimization in heterogeneous wireless networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 7, pp. 1207–1222, Jul. 2012.

[22] R. Atawia and H. Gacanin, "Self-deployment of future indoor Wi-Fi networks: An artificial intelligence approach," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017, pp. 1–6.

[23] G. Cao, Z. Lu, X. Wen, T. Lei, and Z. Hu, "AIF: An artificial intelligence framework for smart wireless network management," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 400–403, Feb. 2018.

[24] Z. Han, X. Wen, W. Zheng, Z. Lu, and T. Lei, "Artificial intelligence based handoff management for dense WLANs: A deep learning approach," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–6.

[25] G. F. Luger, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Harlow, U.K.: Pearson Educ., 2005.

[26] A. Ioannou and S. Weber, "A survey of caching policies and forwarding mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2847–2886, 4th Quart., 2016.

[27] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, "Caching in information-centric networking: Strategies, challenges, and future research directions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1443–1474, 2nd Quart., 2018.

[28] 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.

[29] S. Glass, I. Mahgoub, and M. Rathod, "Leveraging MANET-based cooperative cache discovery techniques in VANETs: A survey and analysis," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2640–2661, 4th Quart., 2017.

[30] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

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

[32] N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 4th Quart., 2019.

[33] 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.

[34] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of green information-centric networking: Research issues and challenges." *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1455–1472, 3rd Quart., 2015.

[35] D. K. Krishnappa, M. Zink, C. Griwodz, and P. Halvorsen, "Cache-centric video recommendation: An approach to improve the efficiency of You Tube caches," *ACM Trans. Multimedia Comput. Commun.*, vol. 11, no. 4, pp. 1–20, 2015.

[36] C.-F. Kao and C.-N. Lee, "Aggregate profit-based caching replacement algorithms for streaming media transcoding proxy systems," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 221–230, Feb. 2007.

[37] P. Pahlevani *et al.*, "Novel concepts for device-to-device communication using network coding," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 32–39, Apr. 2014.

[38] D. Xu, A. Samanta, Y. Li, M. Ahmed, J. Li, and P. Hui, "Network coding for data delivery in caching at edge: Concept, model, and algorithms," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10066–10080, Oct. 2019.

[39] Z. Li, Y. Liu, H. Zhu, and L. Sun, "COFF: Contact-duration-aware cellular traffic offloading over delay tolerant networks," *IEEE Trans. Veh. Technol*, vol. 64, no. 11, pp. 5257–5268, Nov. 2015.

[40] Z. Liu, J. Wang, W. Xu, S. Li, and J. Lin, "Inter-session inter-layer network coding-based dual distributed control for heterogeneous-service networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2014, pp. 3230–3235.

[41] A. Liu and V. K. N. Lau, "Mixed-timescale precoding and cache control in cached MIMO interference network," *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6320–6332, Dec. 2013.

[42] A. Liu and V. K. N. Lau, "Cache-enabled opportunistic cooperative MIMO for video streaming in wireless systems," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 390–402, Jan. 2014.

[43] 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.

[44] M. Tao, E. Chen, H. Zhou, and W. Yu, "Content-centric sparse multicast beamforming for cache-enabled cloud ran," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6118–6131, Sep. 2016.

[45] S. P. Shariatpanahi, H. Shah-Mansouri, and B. H. Khalaj, "Caching gain in wireless networks with fading: A multi-user diversity perspective," in *Proc. IEEE Wireless Commun. Netw. Conf (WCNC)*, Istanbul, Turkey, Apr. 2014, pp. 930–935.

[46] A. Sengupta, R. Tandon, and O. Simeone, "Fog-aided wireless networks for content delivery: Fundamental latency tradeoffs," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6650–6678, Oct. 2017.

[47] T. X. Vu, S. Chatzinotas, and B. Ottersten, "Edge-caching wireless networks: Performance analysis and optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2827–2839, Apr. 2018.

[48] B. Zhou, Y. Cui, and M. Tao, "Optimal dynamic multicast scheduling for cache-enabled content-centric wireless networks," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 2956–2970, Jul. 2017.

[49] A. Ghorbel, M. Kobayashi, and S. Yang, "Cache-enabled broadcast packet erasure channels with state feedback," in *Proc. 53rd Annu. Allerton Conf.*, Sep./Oct. 2015, pp. 1446–1453.

[50] L. Xiang, D. W. K. Ng, T. Islam, R. Schober, V. W. Wong, and J. Wang, "Cross-layer optimization of fast video delivery in cache-and buffer-enabled relaying networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11366–11382, Dec. 2017.

[51] Z. Chen, Y. Liu, B. Zhou, and M. Tao, "Caching incentive design in wireless D2D networks: A Stackelberg game approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[52] F. Gabry, V. Bioglio, and I. Land, "On energy-efficient edge caching in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3288–3298, Dec. 2016.

[53] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, "Match to cache: Joint user association and backhaul allocation in cache-aware small cell networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 3082–3087.

[54] N. Giatsoglou, K. Ntontin, E. Kartsakli, A. Antonopoulos, and C. Verikoukis, "D2D-aware device caching in mmWave-cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2025–2037, Sep. 2017.

[55] 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)*, Jun. 2014, pp. 2648–2653.

[56] M. Abrams, C. R. Standridge, G. Abdulla, S. Williams, and E. A. Fox, "Caching proxies: Limitations and potentials," Comput. Sci. Dept., Virginia Tech., Blacksburg, VA, USA, Rep. TR-95-010, 1995.

[57] M. Arlitt, L. Cherkasova, J. Dilley, R. Friedrich, and T. Jin, "Evaluating content management techniques for Web proxy caches," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 27, no. 4, pp. 3–11, 2000.

[58] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 1999, pp. 126–134.

[59] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, San Diego, CA, USA, Aug. 2007, pp. 1–13.

[60] R. Wang, X. Peng, J. Zhang, and K. B. Letaief, "Mobility-aware caching for content-centric wireless networks: Modeling and methodology," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 77–83, Aug. 2016.

[61] B. Chen, C. Yang, and Z. Xiong, "Optimal caching and scheduling for cache-enabled D2D communications," *IEEE Commun. Lett.*, vol. 21, no. 5, pp. 1155–1158, May 2017.

[62] N. Golrezaei, P. Mansourifard, A. F. Molisch, and A. G. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3665–3676, Jul. 2014.

[63] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Scaling behavior for device-to-device communications with distributed caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 4286–4298, Jul. 2014.

[64] D. Malak and M. Al-Shalash, "Optimal caching for device-to-device content distribution in 5G networks," in *IEEE Globecom Workshops (GC Wkshps)*, Austin, TX, USA, Dec. 2014, pp. 863–868.

[65] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Wireless device-to-device communications with distributed caching," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2012, pp. 2781–2785.

[66] M. Afshang, H. S. Dhillon, and P. H. J. Chong, "Fundamentals of cluster-centric content placement in cache-enabled device-to-device networks," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2511–2526, Apr. 2016.

[67] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.

[68] M. Ji, G. Caire, and A. F. Molisch, "The throughput-outage tradeoff of wireless one-hop caching networks," *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6833–6859, Dec. 2015.

[69] M. Taghizadeh, K. Micinski, S. Biswas, C. Ofria, and E. Torng, "Distributed cooperative caching in social wireless networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 6, pp. 1037–1053, Jun. 2013.

[70] S. Gitzenis, G. S. Paschos, and L. Tassiulas, "Asymptotic laws for joint content replication and delivery in wireless networks," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 2760–2776, May 2013.

[71] K. Zhu, W. Zhi, L. Zhang, X. Chen, and X. Fu, "Social-aware incentivized caching for D2D communications," *IEEE Access*, vol. 4, pp. 7585–7593, 2016.

[72] E. Bastug, M. Bennis, and M. Debbah, "Social and spatial proactive caching for mobile data offloading," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 581–586.

[73] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3212–3229, Jun. 2016.

[74] J. Liu, B. Bai, J. Zhang, and K. B. Letaief, "Cache placement in fog-RANs: From centralized to distributed algorithms," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7039–7051, Nov. 2017.

[75] M. Sheng, C. Xu, J. Liu, J. Song, X. Ma, and J. Li, "Enhancement for content delivery with proximity communications in caching enabled wireless networks: Architecture and challenges," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 70–76, Aug. 2016.

[76] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[77] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 131–139, Feb. 2014.

[78] A. Gharaibeh, A. Khreishah, B. Ji, and M. Ayyash, "A provably efficient online collaborative caching algorithm for multicell-coordinated systems," *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 1863–1876, Aug. 2016.

[79] D. Liu and C. Yang, "Energy efficiency of downlink networks with caching at base stations," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 907–922, Apr. 2016.

[80] J. Nash, "Non-cooperative games," *Ann. Math.*, vol. 54, pp. 286–295, Sep. 1951.

[81] A. Antonopoulos, E. Kartsakli, and C. Verikoukis, "Game theoretic D2D content dissemination in 4G cellular networks," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 125–132, Jun. 2014.

[82] E. Baştuğ *et al.*, "Big data meets TELCOS: A proactive caching perspective," *J. Commun. Netw.*, vol. 17, no. 6, pp. 549–557, Dec. 2015.

[83] M. A. Kader *et al.*, "Leveraging big data analytics for cache-enabled wireless networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, San Diego, CA, USA, Dec. 2015, pp. 1–6.
[84] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," in *Proc. Emerg. Artif. Intell. Appl. Comput. Eng.*, vol. 160, 2007, p. 324.
[85] D. Heckerman, "A tutorial on learning with Bayesian networks," in *Learning in Graphical Models*. Heidelberg, Germany: Springer, 1998, pp. 301–354.
[86] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning* (Springer Series in Statistics), vol. 1. New York, NY, USA: Springer, 2001.
[87] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning (Adaptive Computation and Machine Learning Series)*. Cambridge, MA, USA: MIT Press, 2006.
[88] A. Quintero and O. Garcia, "A profile-based strategy for managing user mobility in third-generation mobile systems," *IEEE Commun. Mag.*, vol. 42, no. 9, pp. 134–139, Sep. 2004.
[89] K. Majumdar and N. Das, "Mobile user tracking using a hybrid neural network," *Wireless Netw.*, vol. 11, no. 3, pp. 275–284, 2005.
[90] C. Yu *et al.*, "Modeling user activity patterns for next-place prediction," *IEEE Syst. J.*, vol. 11, no. 2, pp. 1060–1071, Jun. 2017.
[91] X. Chen, F. Mériaux, and S. Valentin, "Predicting a user's next cell with supervised learning based on channel states," in *Proc. IEEE Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2013, pp. 36–40.
[92] P. Sandhir and K. Mitchell, "A neural network demand prediction scheme for resource allocation in cellular wireless systems," in *Proc. IEEE Region 5 Conf.*, 2008, pp. 1–6.
[93] P. Fazio, F. De Rango, and I. Selvaggi, "A novel passive bandwidth reservation algorithm based on neural networks path prediction in wireless environments," in *Proc. Int. Symp. Perform. Eval. Comput. Telecommun. Syst. (SPECTS)*, Jul. 2010, pp. 38–43.
[94] A. Adeel, H. Larijani, A. Javed, and A. Ahmadinia, "Critical analysis of learning algorithms in random neural network based cognitive engine for LTE systems," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, Glasgow, U.K., 2015, pp. 1–5.
[95] R. Narasimhan and D. C. Cox, "A handoff algorithm for wireless systems using pattern recognition," in *Proc. 9th IEEE Int. Symp. Pers. Indoor Mobile Radio Commun.*, Boston, MA, USA, Sep. 1998, pp. 335–339.
[96] P. Bhattacharya, A. Sarkar, I. Sarkar, and S. Chatterjee, "An ANN based call handoff management scheme for mobile cellular network," *Int. J. Wireless Mobile Netw.*, vol. 5, no. 6, pp. 125–135, Dec. 2013.
[97] V. N. Vapnic, *Statistical Learning Theory*. New York, NY, USA: Wiley Intersci., 1998.
[98] S. Akoush and A. Sameh, "The use of Bayesian learning of neural networks for mobile user position prediction," in *Proc. 7th Int. Conf. Intell. Syst. Design Appl. (ISDA)*, Oct. 2007, pp. 441–446.
[99] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.
[100] L. Lei, L. You, G. Dai, T. X. Vu, D. Yuan, and S. Chatzinotas, "A deep learning approach for optimizing content delivering in cache-enabled HetNet," in *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, Bologna, Italy, Aug. 2017, pp. 449–453.
[101] Y. Bengio and Y. LeCun, "Scaling learning algorithms towards AI," *Large Scale Kernel Mach.*, vol. 34, no. 5, pp. 1–41, 2007.
[102] S. Haykin, "A comprehensive foundation," *Neural Netw.*, vol. 2, p. 41, Oct. 2004.
[103] M. A. Hearst, "Support vector machines," *IEEE Intell. Syst.*, vol. 13, no. 4, pp. 18–28, Jul. 1998, doi: 10.1109/5254.708428.
[104] B. Schölkopf, A. J. Smola, and F. Bach, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.
[105] L. Wang, *Support Vector Machines: Theory and Applications*, vol. 177. New York, NY, USA: Springer, 2005.
[106] M. S. ElBamby, M. Bennis, W. Saad, and M. Latva-Aho, "Content-aware user clustering and caching in wireless small cell networks," in *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2014, pp. 945–949.
[107] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.
[108] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient *k*-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.
[109] J. Wu, *Advances in K-Means Clustering: A Data Mining Thinking*. New York, NY, USA: Springer, 2012.
[110] P.-N. Tan, M. Steinbach, and V. Kumar, "Data mining cluster analysis: Basic concepts and algorithms," in *Introduction to Data Mining*. Piscataway, NJ, USA: IEEE Press, 2013.
[111] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.
[112] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[113] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[114] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.
[115] C. Li *et al.*, "Detection and defense of DDoS attack-based on deep learning in OpenFlow-based SDN," *Int. J. Commun. Syst.*, vol. 31, no. 5, 2018, Art. no. e3497.
[116] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[117] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 4520–4524.
[118] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, vol. 135. Cambridge, MA, USA: MIT Press, 1998.
[119] S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, "A game theoretic approach for distributed resource allocation and orchestration of softwarized networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 721–735, Mar. 2017.
[120] A. Nowé, P. Vrancx, and Y.-M. De Hauwere, "Game theory and multi-agent reinforcement learning," in *Reinforcement Learning*. Berlin, Germany: Springer, 2012, pp. 441–470.
[121] J.-F. Xie, R.-C. Xie, T. Huang, J. Liu, F. R. Yu, and Y.-J. Liu, "Caching resource sharing in radio access networks: A game theoretic approach," *Front. Inf. Technol. Electron. Eng.*, vol. 17, no. 12, pp. 1253–1265, 2016.
[122] V. Mnih and K. Kavukcuoglu, "Methods and apparatus for reinforcement learning," U.S. Patent 9 679 258, Jun. 13, 2017.
[123] B. Bharath, K. G. Nagananda, D. Gündüz, and H. V. Poor, "Learning-based content caching with time-varying popularity profiles," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017, pp. 1–6.
[124] 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.
[125] G. Shen, L. Pei, P. Zhiwen, L. Nan, and Y. Xiaohu, "Machine learning based small cell cache strategy for ultra dense networks," in *Proc. 9th IEEE Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2017, pp. 1–6.
[126] S. Rathore, J. H. Ryu, P. K. Sharma, and J. H. Park, "DeepCachNet: A proactive caching framework based on deep learning in cellular networks," *IEEE Netw.*, vol. 33, no. 3, pp. 130–138, May 2019.
[127] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo state networks for proactive caching and content prediction in cloud radio access networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Washington, DC, USA, Dec. 2016, pp. 1–6.
[128] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo state networks for proactive caching in cloud-based radio access networks with mobile users," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3520–3535, Jun. 2017.
[129] P. Blasco and D. Gündüz, "Learning-based optimization of cache content in a small cell base station," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 1897–1903.
[130] P. Blasco and D. Gunduz, "Multi-armed bandit optimization of cache content in wireless infostation networks," in *Proc. Int. Symp. Inf. Theory (ISIT)*, Jun. 2014, pp. 51–55.
[131] 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.
[132] B. B. Nagaraja and K. G. Nagananda, "Caching with unknown popularity profiles in small cell networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 1–6.
[133] E. Baştuğ, M. Bennis, and M. Debbah, "A transfer learning approach for cache-enabled wireless networks," in *Proc. 13th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, Mumbai, India, May 2015, pp. 161–166.

[134] Y. Jiang, M. Ma, M. Bennis, F. Zheng, and X. You, "A novel caching policy with content popularity prediction and user preference learning in fog-RAN," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2017, pp. 1–6.

[135] J. Yin, L. Li, H. Zhang, X. Li, A. Gao, and Z. Han, "A prediction-based coordination caching scheme for content centric networking," in *Proc. 27th Wireless Opt. Commun. Conf. (WOCC)*, Hualien, Taiwan, Apr. 2018, pp. 1–5.

[136] L. Ale, N. Zhang, H. Wu, D. Chen, and T. Han, "Online proactive caching in mobile edge computing using bidirectional deep recurrent neural network," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5520–5530, Jun. 2019.

[137] N. Zhang, K. Zheng, and M. Tao, "Using grouped linear prediction and accelerated reinforcement learning for online content caching," in *IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Kansas City, MO, USA, May 2018, pp. 1–6.

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

[139] B. Shahriari, M. Moh, and T.-S. Moh, "Generic online learning for partial visible dynamic environment with delayed feedback: Online learning for 5G C-RAN load-balancer," in *Proc. Int. Conf. High Perform. Comput. Simulat. (HPCS)*, Genoa, Italy, Jul. 2017, pp. 176–185.

[140] X. Lin *et al.*, "Marl-based distributed cache placement for wireless networks," *IEEE Access*, vol. 7, pp. 62606–62615, 2019.

[141] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Smart caching in wireless small cell networks via contextual multi-armed bandits," in *IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–7.

[142] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Context-aware proactive content caching with service differentiation in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 1024–1036, Feb 2017.

[143] 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)*, Barcelona, Spain, 2014, pp. 917–921.

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

[145] H. Zhu, Y. Cao, X. Wei, W. Wang, T. Jiang, and S. Jin, "Caching transient data for Internet of Things: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2074–2083, Apr. 2019.

[146] K. N. Doan, M. Vaezi, W. Shin, H. V. Poor, H. Shin, and T. Q. S. Quek, "Power allocation in cache-aided NOMA systems: Optimization and deep reinforcement learning approaches," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 630–644, Jan. 2020.

[147] F. Jiang, Z. Yuan, C. Sun, and J. Wang, "Deep Q-learning-based content caching with update strategy for fog radio access networks," *IEEE Access*, vol. 7, pp. 97505–97514, 2019.

[148] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[149] 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.

[150] S. O. Somuyiwa, A. György, and D. Gündüz, "A reinforcement-learning approach to proactive caching in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1331–1344, Jun. 2018.

[151] Y. He, Z. Zhang, and Y. Zhang, "A big data deep reinforcement learning approach to next generation green wireless networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017, pp. 1–6.

[152] Y. He *et al.*, "Deep reinforcement learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10433–10445, Nov. 2017.

[153] W. Li, J. Wang, G. Zhang, L. Li, Z. Dang, and S. Li, "A reinforcement learning based smart cache strategy for cache-aided ultra-dense network," *IEEE Access*, vol. 7, pp. 39390–39401, 2019.

[154] Y. Sun, M. Peng, and S. Mao, "Deep reinforcement learning-based mode selection and resource management for green fog radio access networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1960–1971, Apr. 2019.

[155] C. Ma, Z. Lin, L. Marini, J. Li, and B. Vucetic, "Learning automaton based distributed caching for mobile social networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2016, pp. 1–6.

[156] S. M. S. Tanzil, W. Hoiles, and V. Krishnamurthy, "Adaptive scheme for caching YouTube content in a cellular network: Machine learning approach," *IEEE Access*, vol. 5, pp. 5870–5881, 2017.

[157] K. C. Tsai, L. Wang, and Z. Han, "Mobile social media networks caching with convolutional neural network," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2018, pp. 83–88.

[158] H. Hao, C. Xu, M. Wang, H. Xie, Y. Liu, and D. O. Wu, "Knowledge-centric proactive edge caching over mobile content distribution network," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Honolulu, HI, USA, Apr. 2018, pp. 450–455.

[159] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, D. Niyato, and D. I. Kim, "Distributed deep learning at the edge: A novel proactive and cooperative caching framework for mobile edge networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1220–1223, Aug. 2019.

[160] 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 (INFOCOM WKSHPS)*, Apr. 2018, pp. 511–516.

[161] 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.

[162] 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.

[163] 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)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[164] B. Bharath, K. G. Nagananda, and H. V. Poor, "A learning-based approach to caching in heterogenous small cell networks," *IEEE Trans. Commun.*, vol. 64, no. 4, pp. 1674–1686, Apr. 2016.

[165] P. Cheng *et al.*, "Localized small cell caching: A machine learning approach based on rating data," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1663–1676, Feb. 2019.

[166] T. Gao, M. Chen, H. Gu, and C. Yin, "Reinforcement learning based resource allocation in cache-enabled small cell networks with mobile users," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Qingdao, China, Oct. 2017, pp. 1–6.

[167] Y. He, C. Liang, F. R. Yu, N. Zhao, and H. Yin, "Optimization of cache-enabled opportunistic interference alignment wireless networks: A big data deep reinforcement learning approach," in *Proc. IEEE Int. Conf. Commun (ICC)*, Paris, France, May 2017, pp. 1–6.

[168] W.-X. Liu, J. Zhang, Z.-W. Liang, L.-X. Peng, and J. Cai, "Content popularity prediction and caching for ICN: A deep learning approach with SDN," *IEEE Access*, vol. 6, pp. 5075–5089, 2018.

[169] W. Wang, R. Lan, J. Gu, A. Huang, H. Shan, and Z. Zhang, "Edge caching at base stations with device-to-device offloading," *IEEE Access*, vol. 5, pp. 6399–6410, 2017.

[170] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, "Optimal and scalable caching for 5G using reinforcement learning of space-time popularities," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 180–190, Feb. 2018.

[171] 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.

[172] 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.

[173] F. Alotaibi, A. Eryilmaz, and H. El Gamal, "Dynamic pricing and proactive caching with unknown demand profile," in *Proc. 54th Annu. Allerton Conf. Commun. Control Comput. (Allerton)*, Monticello, IL, USA, Sep. 2016, pp. 711–716.

[174] Y. Guo, F. R. Yu, J. An, K. Yang, Y. He, and V. C. M. Leung, "Buffer-aware streaming in small-scale wireless networks: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6891–6902, Jul. 2019.

[175] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, "Green resource allocation based on deep reinforcement learning in content-centric IoT," *IEEE Trans. Emerg. Topics Comput.*, early access, Feb. 13, 2018, doi: 10.1109/TETC.2018.2805718.

[176] T. Hou, G. Feng, S. Qin, and W. Jiang, "Proactive content caching by exploiting transfer learning for mobile edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017, pp. 1–6.

[177] Y. Li, T. Wu, P. Hui, D. Jin, and S. Chen, "Social-aware D2D communications: Qualitative insights and quantitative analysis," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 150–158, Jun. 2014.

Muhammad Sheraz received the B.Sc. degree in electrical engineering from the University of Engineering and Technology, Peshawar, Pakistan, in 2007, and the M.Sc. degree in electrical and electronic engineering from the University of Bradford, U.K., in 2010. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Tsinghua University, Beijing, China. His research interests include wireless cooperative communication, D2D communications, fog and edge computing, machine learning, artificial intelligence in wireless networks, and big data.

Manzoor Ahmed received the B.E. and M.S. degrees in electrical engineering and computer science from Balochistan Engineering University, Khuzdar, Pakistan, in 1996 and 2010, respectively, and the Ph.D. degree in communication and information systems from the Beijing University of Posts and Telecommunications, China, in 2015. From 1997 to 2000, he was a Lecturer with Balochistan Engineering University and a Telecomm Engineer in government-owned telecommunication service provider NTC, Pakistan, from 2000 to 2011. He was a Postdoctoral Researcher from the Electrical Engineering Department, Tsinghua University, China, from 2015 to 2018. He is currently a Faculty Member with the Department of Computer Science and Technology, Qingdao University. His research interests include resource allocation and offloading in vehicular communications and networking, fog and edge computing, socially aware D2D communication, and physical layer security. He has several research publications in IEEE top journals and conferences. He received several awards, including the Distinction Award from the President of Pakistan, the Best Employee Award from NTC, and the Best Paper Award from the 2014 GameNets Conference.

Xueshi Hou received the B.Eng. degree in electronic engineering from Tsinghua University, Beijing, China, in 2015. She is currently pursuing the Ph.D. degree with the University of California at San Diego, La Jolla, CA, USA. Her research interests include multimedia, virtual reality, mobile computing, and wireless communications.

Yong Li (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from Tsinghua University in 2012, where he is currently a Tenured Associate Professor with the Department of Electronic Engineering. His research interests include machine learning and big data mining, particularly, automatic machine learning and spatial-temporal data mining for urban computing, recommender systems, and knowledge graphs. He has served as a general chair, a TPC chair, a SPC/TPC member for several international workshops and conferences, and he is on the editorial board of two IEEE journals. He has published over 100 papers on first-tier international conferences and journals, including KDD, WWW, UbiComp, SIGIR, *Association for the Advancement of Artificial Intelligence*, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and IEEE TRANSACTIONS ON MOBILE COMPUTING, and his papers have total citations more than 8 300. Among them, ten are ESI Highly Cited Papers in computer science, and five receive conference Best Paper (run-up) Awards. He received the IEEE 2016 ComSoc Asia–Pacific Outstanding Young Researchers, the Young Talent Program of China Association for Science and Technology, and the National Youth Talent Support Program.

Depeng Jin (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Tsinghua University in 1995, 1997, and 2000, respectively. From 2000 to 2003, he was an Assistant Professor. From 2004 to 2012, he was an Associate Professor. Since 2012, he has been a Professor with Department of Electronic Engineering, Tsinghua University. His research focuses on communication and networking. His current interests are in the topics of software-defined networks, wireless mobile network, and mobile big data.

Zhu Han (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.

From 2000 to 2002, he was an Research and Development Engineer with JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, Boise, ID, USA. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department as well as in the Computer Science Department, University of Houston, Houston, TX, USA. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. He received the NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the *Journal on Advances in Signal Processing* in 2015, the IEEE Leonard G. Abraham Prize in the field of Communications Systems (Best Paper Award in IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS) in 2016, and several best paper awards in IEEE conferences. He was an IEEE Communications Society Distinguished Lecturer from 2015 to 2018, and is AAAS Fellow since 2019, and a ACM Distinguished Member since 2019. He is 1% highly cited researcher since 2017 according to Web of Science.

Tao Jiang (Fellow, IEEE) received the Ph.D. degree in information and communication engineering from the Huazhong University of Science and Technology, Wuhan, China, in April 2004, where he is currently a Distinguished Professor with the Wuhan National Laboratory for Optoelectronics and School of Electronics Information and Communications. From August 2004 to December 2007, he worked in some universities, such as Brunel University and the University of Michigan–Dearborn. He has authored or coauthored more 300 technical papers in major journals and conferences and nine books/chapters in the areas of communications and networks. He served or is serving as symposium technical program committee membership of some major IEEE conferences, including INFOCOM, GLOBECOM, and ICC. He was invited to serve as a TPC Symposium Chair for the IEEE GLOBECOM 2013, IEEEE WCNC 2013, and ICCC 2013. He is served or serving as an Associate Editor of some technical journals in communications, including in IEEE NETWORK, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE INTERNET OF THINGS JOURNAL, and he is the Associate Editor-in-Chief of China Communications.