IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 September 2022, accepted 2 November 2022, date of publication 7 November 2022,
date of current version 21 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3220694

SURVEY

A Survey on Mobile Edge Computing for Video Streaming: Opportunities and Challenges

MUHAMMAD ASIF KHAN[1], (Senior Member, IEEE), EMNA BACCOUR[2], ZINA CHKIRBENE[1], AIMAN ERBAD[2], (Senior Member, IEEE), RIDHA HAMILA[1], (Senior Member, IEEE), MOUNIR HAMDI[2], (Fellow, IEEE), AND MONCEF GABBOUJ[3], (Fellow, IEEE)
[1]Department of Electrical Engineering, Qatar University, Doha, Qatar
[2]Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Qatar Foundation, Doha, Qatar
[3]Department of Computing Sciences, Tampere University, 33101 Tampere, Finland

Corresponding author: Muhammad Asif Khan (mkhan@qu.edu.qa)

This paper was made possible by Qatar University Internal Grant No. IRCC-2020-001 and PDRA award PDRA7-0606-21012 from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors. Qatar National Library funded the publications of this article.

ABSTRACT 5G communication brings substantial improvements in the quality of service provided to various applications by achieving higher throughput and lower latency. However, interactive multimedia applications (e.g., ultra high definition video conferencing, 3D and multiview video streaming, crowd-sourced video streaming, cloud gaming, virtual and augmented reality) are becoming more ambitious with high volume and low latency video streams putting strict demands on the already congested networks. Mobile Edge Computing (MEC) is an emerging paradigm that extends cloud computing capabilities to the edge of the network i.e., at the base station level. To meet the latency requirements and avoid the end-to-end communication with remote cloud data centers, MEC allows to store and process video content (e.g., caching, transcoding, pre-processing) at the base stations. Both video on demand and live video streaming can utilize MEC to improve existing services and develop novel use cases, such as video analytics, and targeted advertisements. MEC is expected to reshape the future of video streaming by providing ultra-reliable and low latency streaming (e.g., in augmented reality, virtual reality, and autonomous vehicles), pervasive computing (e.g., in real-time video analytics), and blockchain-enabled architecture for secure live streaming. This paper presents a comprehensive survey of recent developments in MEC-enabled video streaming bringing unprecedented improvement to enable novel use cases. A detailed review of the state-of-the-art is presented covering novel caching schemes, optimal computation offloading, cooperative caching and offloading and the use of artificial intelligence (i.e., machine learning, deep learning, and reinforcement learning) in MEC-assisted video streaming services.

INDEX TERMS Live streaming, machine learning, mobile edge computing, VoD, video Streaming.

I. INTRODUCTION

The emergence of 5G brings substantial improvements in quality of service by achieving higher throughput and lower latency. These advantages enable network providers to tailor the quality of experiences for new use cases across different vertical markets. However, these new capabilities on one side are promising to transform industries, but on the other side, the massive wave of new devices and bandwidth intensive multimedia applications supported by 5G, trigger new challenges. In the context of video streaming, the improved cellular bandwidth has enabled novel use cases of video streaming e.g. ultra HD (e.g. 4K and 8K) videos, 3D videos, 360° videos, virtual and augmented reality and interactive video streaming. These new classes of video streaming put forward new demands in terms of reliable computation and reduced delay to match the desired quality of experience. These services traditionally rely on cloud computing in which the data storage and computational resources reside at the cloud servers. However, as the network scales, the increasing number of users requests can lead to significant increase in latency. Several factors add to the end-to-end latency i.e.,

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

delay caused by offloading the computation to the cloud, delay of processing and queuing at the cloud server and network communication delay to transmit the requested video back to the user. Furthermore, as the number of requests grows, meeting the delay requirement of latency-sensitive applications becomes challenging.

While latency is the key factor in the context of video streaming, there are a number of other benefits that can be achieved using MEC as compared to the traditional cloud computing paradigm. For instance, popular video streaming services such as Netflix [1], Hulu [2], and Amazon Prime [3] create a heavy load on network. MEC can be used to cache popular content closer to the end users for a smoother experience. In cloud-based video streaming, the content is stored relatively far from the user geographical locations leading to a poor user's experience when there is a congestion along the end to end route. Cloud-based streaming services also suffers from quality of service (QoS) degradation when the users requesting similar video content grows abruptly.

To alleviate the issues of traditional cloud service models, mobile edge computing is emerging as a promising solution. The primary benefit of edge computing is to bring the storage and computing capabilities closer to the end users i.e., at the edge of the network. In MEC assisted 5G networks, the storage and computing services can be deployed at the network edge (i.e., within the RAN) to enable network providers better handle latency-sensitive services. MEC can bring a range of benefits to users as well as network providers. By providing storage and computing resources at the edge of the network (i.e. RAN), communication latency is reduced. Content providers can use edge storage capability to cache the content locally which reduces the bandwidth resources on the backhaul links. Distributed computing and storage resources improve resiliency and availability using collaboration among edge servers. Users can access content at lower costs available locally. Edge computing can also enhance security and greater improve scalability due to the fact that any attack on an edge server will affect the users connected to that server and not the whole network. These benefits are of MEC are further detailed in Section III.

MEC is still an emerging area and a huge amount of research work on mobile edge computing is in progress. MEC can help revolutionize several sectors including but not limited to, enterprises, smart buildings, healthcare, vehicles-to-infrastructure (V2I) services, Internet of Things, video streaming and virtual and augmented reality. Several survey papers exist to summarize the ongoing research efforts on mobile edge computing. A large number of these surveys provide a high-level broader view of the research area covering different aspects e.g. architectural development, functional overview, data caching, computational offloading, potential use cases, opportunities and challenges [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. Some of these works cover specific aspects e.g. computation offloading [5], [18], [19], [20], communication [21] and security [22], [23], [24], [25], [26], [27]. There are few studies covering edge computing research in particular applications such as IoT [28], [29], [30], whereas some of these aim to cover the state-of-the-art scientific contributions of integrating edge computing with sophisticated machine learning and deep learning [6], [31], [32], [33], [34]. A summary of these surveys is presented in Table 1.

Although most of the aforementioned survey papers (summarised in Table 1) provide a detailed review of research efforts on mobile edge computing, we realise it is worthy that the huge amount of research works on MEC-assisted video streaming services shall be reviewed and summarized. In this survey, we provide a comprehensive review of the state-of-the-art in mobile edge computing for video streaming use cases. The paper describes different types of video streaming services and the associated challenges, and explains how these applications can benefit from using edge computing. Although we aim at focusing on video streaming, nevertheless we included reference works which are not explicitly proposed for video streaming use cases, but these have good correlation and are readily applicable in the relevant use-cases. We dedicate separate sections for novel research contributions in the area of cooperative device-to-device (D2D) communication and machine learning and their respective benefits and applications in MEC-assisted video streaming.

The organization of the paper is illustrated in Figure 1. Section II presents an overview of video streaming services outlining different types of video streaming services, listing streaming protocols and video streaming challenges in terms of the quality of experience. Section III provides a brief overview of mobile edge computing, explaining the architecture, functional overview, fundamental definitions and related concepts. Section IV explains how MEC can improve video streaming services in different streaming use cases. A detailed review of the state-of-the-art on mobile edge computing for video streaming is presented in Section V, with separate subsections dedicated for research contributions in edge caching and computational offloading. Sections VI and VII summarise the most recent developments on edge computing techniques using cooperative networking and machine learning respectively. The abbreviations and acronyms used in this paper are listed in Table 2.

II. VIDEO STREAMING - TYPES, PROTOCOLS AND CHALLENGES

Video streaming is the method of viewing videos without downloading the media files.

A. TYPES OF VIDEO STREAMING

Video streaming is not only at the forefront of entertainment industry but also is transforming several other sectors such as enterprises, education, retail, tourism, transportation and healthcare. The unprecedented use cases of video streaming are reshaping the Internet, while at the same time, network and content providers are doing huge investments in improving user experience in video streaming services. Generally,

TABLE 1. Comparative study of this paper with existing surveys on edge computing. The symbol ✓ indicates a publication is in the scope of a domain; ✗ marks papers that do not directly cover that area, but from which readers may retrieve some related insights; □ indicates the topic is covered but in less depth.

Ref	One-sentence summary	Scope of the Publication				
		Architecture	Caching	Offloading	ML/DL	Video streaming
Wang *et al.* [31]	Deep learning in edge computing	✗	✗	✗	✓	✗
Deng *et al.* [32]	Artificial Intelligence (AI) in edge computing	✗	✗	✗	✓	✗
Carvalho *et al.* [33]	Using AI for computation offloading in edge computing for	✓	✗	✓	✓	✗
Cao *et al.* [4]	Overview of edge computing research	✓	✗	✗	✗	✗
Lin *et al.* [5]	A survey of computation offloading techniques in edge computing	✓	✗	✓	✓	✗
Shakarami *et al.* [6]	A survey of computation offloading techniques in edge and cloud computing	✓	✗	✓	✗	✗
Xiao *et al.* [22]	A survey on security challenges in edge computing	✓	✗	✗	✗	✗
Yang *et al.* [23]	A survey on integration of bloackchains and edge computing	✓	✗	✗	✗	✗
Liu *et al.* [24]	A survey on secured edge-based data analytics	✓	✗	✗	✗	✗
Jiang *et al.* [19]	A survey of computation offloading in edge computing	✓	✗	✓	✗	✗
Lin *et al.* [20]	A survey of computation offloading in edge computing	✓	✗	✓	✗	✗
Donno *et al.* [7]	A tutorial on modern computing paradigms	✓	✗	✗	✗	✗
Khan *et al.* [8]	A survey on edge computing applications	✓	✗	✓	✗	✗
Hassan *et al.* [9]	A review of edge computing in 5G networks	✗	✗	✓	✗	✗
Chen *et al.* [34]	A review of deep learning applications in edge computing	✗	✗	✓	✓	✗
Zhang *et al.* [25]	A survey on security and privacy in edge computing and	✓	✗	✗	✓	✗
Marjanovic *et al.* [28]	A tutorial on edge computing for crowdsensing	✓	✗	✗	✗	✗
Porambage *et al.* [29]	A survey on application of edge computing in Internet of Things (IoT)	✓	✗	✗	✗	✗
Abbas *et al.* [10]	A tutorial and survey on mobile edge computing	✓	✗	✗	✗	✗
Wang *et al.* [11]	A survey on edge computing services	✓	✓	✓	✗	✗
Baktir *et al.* [12]	A survey on software-defined networking in edge computing	✓	✗	✗	✗	✗
Ahmed *et al.* [13]	A survey of the state-of-the-art in mobile edge computing	✗	✓	✗	✗	✗
Taleb *et al.* [14]	A survey of edge computing in 5G networks	✓	✓	✓	✗	✗
Mao *et al.* [15]	A review of the state-of-the-art in mobile edge computing	✓	✗	✓	✗	✗
Mao *et al.* [21]	A survey of communication techniques in mobile edge computing	✗	✗	✗	✓	✗
Shirazi *et al.* [27]	A tutorial on security in mobile edge and fog computing	✓	✗	✗	✗	✗
Yu *et al.* [30]	A survey and tutorial on edge computing in IoT	✓	✗	✓	✗	✗
Ahmed *et al.* [16]	A survey of mobile edge computing systems	✓	✗	✗	✗	✗
Shi *et al.* [17]	A futuristic overview of mobile edge computing	✓	✗	✓	✗	✗
Jiang *et al.* [35]	MEC in Video Streaming	✓	□	✓	✗	✓
Kanai *et al.* [36]	MEC for multimedia applications	✗	□	□	✗	✗
Zhang *et al.* [37]	MEC for video analytics in public safety	□	✗	✗	✗	□
This survey	A survey of MEC in video streaming covering the MEC architecture, video streaming applications and novel services, and traditional and state-of-the-art techniques for edge-based caching and processing. It also covers the use of ML and pervasive computing including D2D for the next generation intelligent edge services.	✓	✓	✓	✓	✓

TABLE 2. Summary of acronyms.

Acronym	Definition
3D	Three Dimensional
3GPP	Third Generation Partnership Project
4K	Horizontal video resolution of 4000 pixels
5G	Fifth-Generation
ABR	Adaprive Bit Rate
AI	Artificial Intelligence Internet
AR	Augmented Reality
BS	Base Station
CC	Cloud Computing
CDN	Content Delivery Network
CMAF	Common Media Application Format
CNN	Convolutional Neural Network Recurrent
D2D	Device to Device
DL	Deep Learning
DNN	Deep Neural Network
DOS	Denial Of Service
DQN	Deep Q-learning Network
DRL	Deep Reinforcement Learning
EC	Edge Computing
ETSI	European Telecommunication Standards Institute
FIFO	First In First Out
FOV	Field of View
HD	High Definition
HDS	HTTP Dynamic Streaming
HLS	HTTP Live Streaming
HTTP	Hypertext Transfer Protocol
ILP	Integer Linear Programming
INLP	Integer Non-Linear Programming
IoT	Internet of Things
ISG	Industry Specification Group
ISG	Industry Specific Groups
LFU	Least Frequently Used
LLR	Least Likely Requested
LRU	Least Recently Used
LRFU	Least Recently Frequency Used
LSTM	Long-short Term Memory model
MDP	Markov Decision Problem
MEC	Mobile Edge Computing
MEC	Mobile (or Multi-Access) Edge Computing
MILP	Mixed Integer Linear Programming
ML	Machine Learning
MLR	Most Likely Requested
DASH	Dynamic Adaptive Streaming over HTTP
MPV	Most Popular Videos
MPEG	Motion Picture Expert Group
MSS	Microsoft Smooth Streaming
NFV	Network Function Virtualization
PcP	Proactive Cache Policy
P-UPP	Proactive User Preference Profile
PoP	Point of Presence
QoE	Quality of Experience
QoS	Quality of Service
R-UPP	Reactive User Preference Profile
RAN	Radio Access Network
RFC	Request For Comments
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RTMP	Real-Time Messaging Protocol
RTP	Real-Time Transport Protocol
RTSP	Real-Time Streaming Protocol
SIP	Session Initiation Protocol
SRT	Secure Reliable Transport
TDMA	Time Devision Multiple Access
UA	User Attrition
UAV	Unmanned Aerial Vehicle
UPP	User Preference Profile
URRLC	Ultra-reliable Low-latency Communications
V2I	Vehicle to Infrastructure
VoD	Video On Demand
VR	Virtual Reality
WebRTC	Web Real-Time Communications

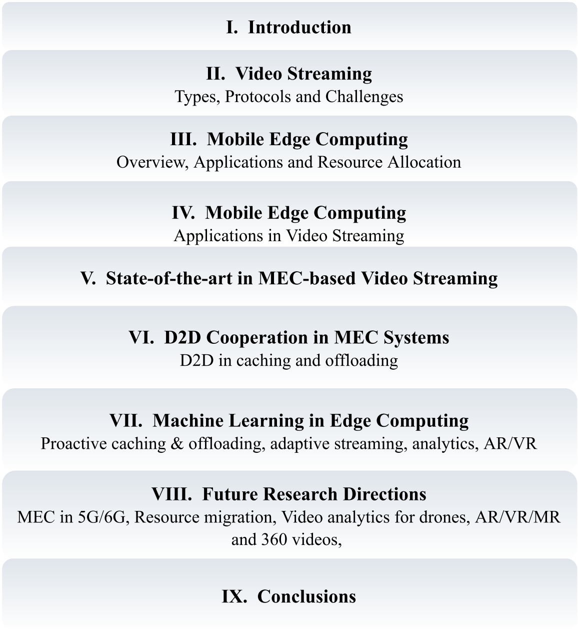
I. Introduction
II. Video Streaming
Types, Protocols and Challenges
III. Mobile Edge Computing
Overview, Applications and Resource Allocation
IV. Mobile Edge Computing
Applications in Video Streaming
V. State-of-the-art in MEC-based Video Streaming
VI. D2D Cooperation in MEC Systems
D2D in caching and offloading
VII. Machine Learning in Edge Computing
Proactive caching & offloading, adaptive streaming, analytics, AR/VR
VIII. Future Research Directions
MEC in 5G/6G, Resource migration, Video analytics for drones, AR/VR/MR and 360 videos,
IX. Conclusions

FIGURE 1. Paper organization: illustration of various topics covered in each section.

video streaming applications can be broadly categorized as Video on Demand (VoD) and Live video streaming.

1) VoD STREAMING

VoD streaming allows users to watch stored videos from any Internet-connected devices at any suitable time. In VoD streaming, content can be prefetched, stored and edited before it is distributed. Popular applications of VoD streaming are Netflix [1], Apple iTunes Store [38], and YouTube. There are three popular models of VoD streaming in today's video streaming world.

2) LIVE STREAMING

Unlike VoD streaming, in live video streaming the video is distributed in real-time directly from the origin device to the destination, without first storing it on a server. Live videos are more sensitive to network delays as compared to VoD streaming. The use of live video streaming is increasing over time [39] due to multiple reasons such as the availability of high quality cameras in the modern smartphones that enable users to shoot high quality videos, and the high data rates of 5G, enabling the users to share videos over the internet in real time. Live video streaming has many attractive applications such as E-Sports and Game Streaming [40], [41], [42], [43], [44], virtual reality and augmented reality, user-generated live streaming (e.g. Periscope [45], Youtube Live [46], Twitch [47], Facebook Live [48], Instagram Live [49], Twitter Live [50] and Ustream [51]) and online learning (e.g. Dacast [52], IBM Cloud Video [51], Kaltura [53], Vimeo Live [54] and Panopto [55]). In the following, we categorize the live streaming by application scenarios:

- Conventional live streaming videos: This type of applications, including Twitch, YouTube live, Facebook live, allows random users to capture live videos using their handheld devices. In such applications, the live system needs to handle a huge volume of videos from broadcasters to thousands of viewers located all over the world in a very short end-to-end latency (e.g., 100 ms [56]). Moreover, to provide the required formats to viewers based on their preferences, the capability of their devices, and the quality of the network, the videos should be transcoded. This transcoding task needs to be done in real-time and adapts on the fly to the available computing resources to match the spontaneous broadcasts.
- Streaming 360° (or panoramic) videos: This type of streaming is more challenging than conventional videos as it requires higher bandwidth availability due to the volume of the content. Moreover, such live stream allows a flexible interaction, i.e., the user can potentially move and expects to see different views of the panorama. Therefore, the latency and delay variance are even stricter in order to update the display without any motion sickness. This delay is called motion-to-photon latency and should not exceed few milliseconds, in order to provide a smooth experience [57].
- VR/AR live streaming: The Virtual Reality (VR) is a simulated experience that can be fictional or similar to the real world. Multiple features characterize the VR systems, which are the imagination, interaction, and immersion, in addition to the integration of the VR user in the virtual environment. The VR experience requires a device with a screen, computing components and sensors that track the user's motions. The VR live streaming has the same requirements as conventional and 360° live videos, including those related to network parameters (e.g., bandwidth, latency, stalling, buffering time, and bitrate switching) and the application parameters (e.g., video quality, frame rate, and resolution) with higher constraints on the delay variance and image freezing to avoid motion sickness. Besides these requirements, the VR system depends also on physical environment factors (e.g., sounds, objects locations, and lighting) and user's profile (e.g., gender and length), which highly influence the quality of the experience [58]. The Augmented Reality (AR) is also a real-world interactive experience, where real or fictive objects are enhanced and integrated into the user's experience. In terms of system requirements, combining the real world environment with the augmented objects is very critical, as the AR algorithm has to superimpose these objects into the coordinates of the user's location with a high accuracy.
  Even though the AR and VR technologies brought new opportunities in various industries and applications (e.g., medicine, maintenance, sports, teaching, tourism and architecture.), they also added new challenges related to image display, delivery and content storage.

Figure 2 illustrates a generic functional architecture of VoD streaming and live streaming. In VoD streaming (black lines), the videos are recorded first, then transcoded to one or more bit rates versions, and then stored on a cloud server. The viewers can request and access these stored videos anytime using different protocols such as HLS. In most of the video deployments, there are also geographically distributed servers (also known as point of presence (POP)) that cache the videos from the remote cloud and serve them to the user when requested. In the case of live streaming (red lines), videos are transmitted to the end user without storing them first on the cloud. However, videos are still being transcoded in live video streaming, as different users can request different bit rate versions of the video based on the quality of the wireless channel.

Both VoD and Live video streaming are likely to co-exist as both have different applications and use cases. VoD gives users the opportunity to watch videos anytime, anywhere and from any device using the Internet. On the other hand, live streaming provides great convenience to users to share videos in real time without first recording them.

B. VIDEO STREAMING PROTOCOLS

Video streaming protocols can be broadly categorized as (1) Push-based (non-HTTP) and (2) Pull-based (HTTP-based) protocols. Push-based protocols are traditional streaming protocols in which the server and client first establish a connection before transmitting data. These include RTMP (Real-Time Messaging Protocol) [59] and RTSP (Real-Time Streaming Protocol) [60], and SIP (Session Initiation Protocol).

Since 2010, pull-based protocols have been introduced e.g. HLS (HTTP Live Streaming) [61], Low Latency HLS [62], MPEG-DASH (Moving Picture Expert Group -Dynamic Adaptive Streaming over HTTP) [63], CMAF (Common Media Application Format) for DASH [64], MSS (Microsoft Smooth Streaming) [65], Adobe HDS (HTTP Dynamic Streaming) [66], SRT (Secure Reliable Transport) [67] and WebRTC (Web Real-Time Communications) [68].

HLS is the most widely used pull-based streaming protocol [69] supported by many media players, web browsers, devices, and streaming media servers. Low-Latency HLS [62] is an improved version of HLS which provide low latency (up to 2 seconds). The Adobe HDS was the first adaptive bitrate (ABR) protocol. Subsequently, Microsoft developed its own adaptive bitrate protocol for video streaming i.e., MSS in 2008 to deliver on-demand video of the 2008 Summer Olympics. MPEG-DASH is an open standard adaptive bitrate streaming protocol, whereas CMAF was developed by joint collaboration of Microsoft, Apple and MPEG to simplify the streaming services. The most recent addition to this list is the open standard WebRTC framework. Today HLS is the standard platform for Apple whereas MPEG-DASH is the international standard [63] for video streaming. Table 3 presents a summary of various video streaming protocols.

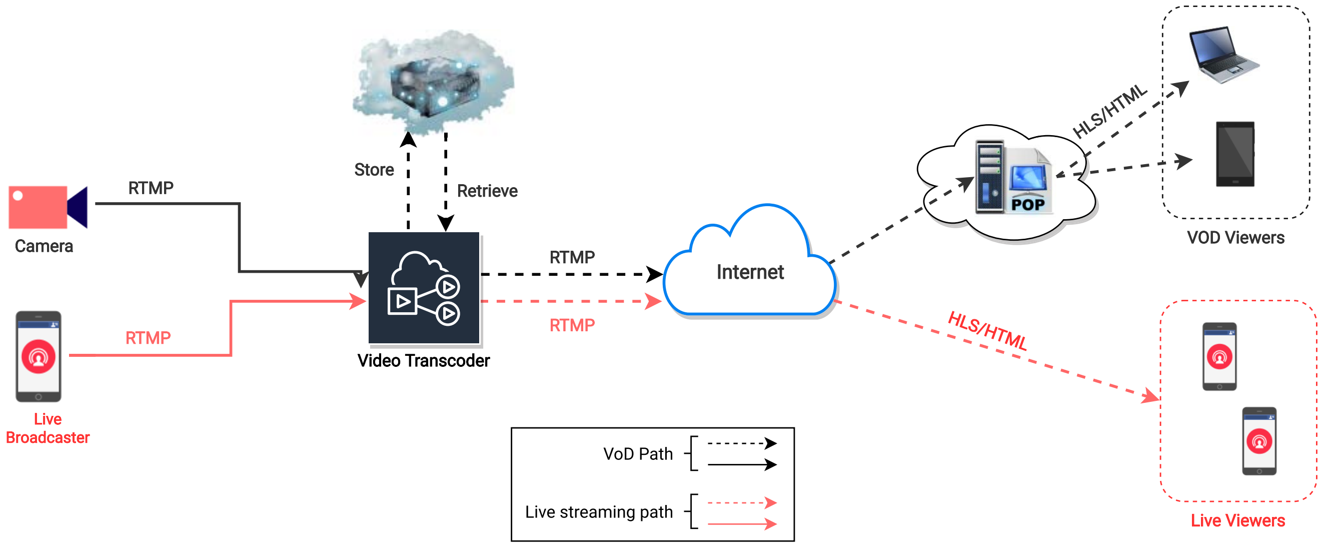

FIGURE 2. Video on Demand (VoD) versus live streaming.

TABLE 3. Video streaming protocols.

Protocol	Developer	Year	Latency(s)	ABR
RTMP	Macromedia	2012	Low	✓
RTSP/RTP	IETF	1998	Ultra low	✗
HLS	Apple	2009	Reduced	✓
LL-HLS	Apple	2009	Low	✓
MPEG-DASH	MPEG	2011	High	✓
CMAF	MPEG	2018	Low	✓
MSS	Microsoft	2008	Reduced	✓
HDS v3	Adobe	2013	Reduced	✓
SRT	Haivision	2013	Low	✓
SIP/RTP	IETF	1999	Reduced	✓
WebRTC	IETF	2011	Ultra low	✓

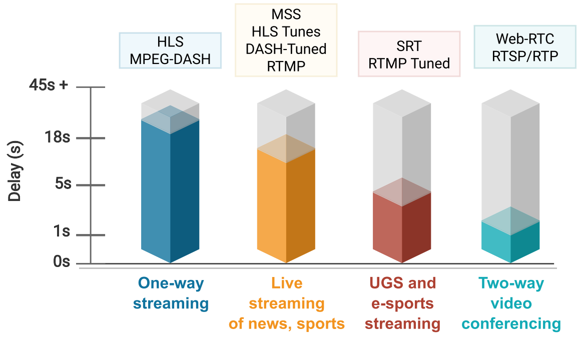

FIGURE 3. Video streaming protocols - delay comparison.

A comparison of streaming delay (i.e. media transfer delay) of these protocols is illustrated in Figure 3.

Video streaming protocols can be generally categorized (based on latency) as (i) High Latency (>18 seconds), (ii) reduced latency (10-18 seconds), (iii) low latency (4-10 seconds) and (iv) Ultra low latency (1-4 seconds) [70].

C. VIDEO STREAMING CHALLENGES

Video streaming applications are gaining popularity due to the inherited convenience for sharing content. However, there are several challenges in live video streaming The three major challenges associated with video streaming services are:

- *Limited Bandwidth:* Modern 5G networks offer higher data rates over cellular links but the growing use of video content with a much higher resolution than before is making the bandwidth limitations more prevalent. In particular, the modern 4K (3840 × 2160), 8K (7680 × 4320), and 360° videos are bandwidth intensive [71]. On average, the 4K videos requires bit rate of 20-50 Mbps, whereas 8K videos requires 50-200 Mbps [72]. To understand the bandwidth requirement of these videos, a 60-frame 4K video consumes 1GB to 10GB of traffic per minute, and a 20 min 4K video requires almost 100 GB of traffic [72]. The use of these high resolution videos is also continuously increasing. It is estimated that by 2023, 66% of connected flat-panel TV sets will be 4K. [73] The 360° videos are even more bandwidth intensive as in these, pixels are transmitted to users from every direction. The efficient transmission of such huge volumes of bandwidth intensive videos is a challenge yet to continue. As also indicated in [74], 5G systems fall short of providing a full immersive extended reality (XR) experience due to lack of support for ultra-low latency and higher data rates in such applications.
- *Latency:* Video streaming in general and live streaming in particular are sensitive to latency. Higher latency leads to poor user experience in many applications, such as real-time online video gaming, virtual reality, and live streaming of high resolution videos such as 4K and 8K. With the existing content distribution architecture that relies on cloud computing, latency guarantees is a critical challenge [21]. Similarly, in the next generation Industrial IoT (IIoT) and applications

such as autonomous vehicles, augmented reality and medical imaging for remote surgeries, 5G cannot meet the sub-millisecond latency [75].

In VoD streaming, videos are stored in remote cloud servers. The larger propagation distance between the cloud servers and the end users encounter delay deteriorate the user experience. To cope with this, content delivery network (CDN) servers are usually deployed to cache content from remote cloud servers to relatively closer servers located in different geographical areas. However, the CDN servers are deployed in relatively few locations and often not located in densely populated areas where viewers may reside. The transmission of video streams from these centralized cloud servers to viewers that are far apart, requires extensive bandwidth and can often create bottlenecks as the number of viewers grows. This results in inconsistent and higher startup delays, video quality degradation, and sometimes the inability to join popular live streams.

In contrast to VoD streaming live video streaming is more sensitive to network delays due to the fact that live videos are not cached and are directly transmitted from origin to the viewer. Live videos hold users' attention $10 - 20\times$ longer than VoD videos [76]. In Facebook, live videos appear at the top of news feed. Also, users in a page receive notifications about a new live video post, thus the number of viewers increases rapidly. Furthermore, live streaming has random viewing patterns with very high peaks. The high number of simultaneous user requests can cause a problem known as "Thundering Herd problem" [77]. A fine example of such sudden increase in demand of live video is the popular online trivia craze [78], where the demand for online streams grew from zero to over one million viewers in just a matter of minutes. Another example is a 45 minutes video of two people exploding a watermelon with rubber bands, which reached a peak of over 800,000 simultaneous viewers [79]. Thus, the rapidly increasing viewing patterns of live videos can cause network congestion that results in higher end to end delay.

- *Jitter:* Jitter or delay variation is the undesired deviation from true periodicity of an assumed periodic signal. Jitters can be caused by fluctuations in queuing and scheduling delays [80]. Packets transmitted on the networks encounter different delays due to two reasons: First, packets route through the network independently, Second, network devices receives packets in queue and thus encounter different queuing delay. Thus, packets transmitted even at almost same time (consecutive packets) experience large variations in end to end delay. Jitter is a considerable issue in video streaming and can degrade QoS [81], [82]. Jitter requirements vary as per application ranging from $10ms$˘ $50ms$ e.g., VoD streaming ($\leq 50ms$) [83], videoconferencing and interactive video streaming ($\leq 30ms$) [84].

III. MOBILE EDGE COMPUTING

In this section, we elaborate the concept of mobile edge computing (MEC) and its key advantages to improve video streaming. First, a brief overview of MEC architecture and its benefits to the network operators and end users are presented. Then, we discuss some relevant works to understand the state-of-the-art in MEC for novel emerging services (not specifically video streaming). The state-of-the-art in MEC for novel video streaming services is detailed in Section V.

A. OVERVIEW AND DEFINITIONS

The term Mobile Edge Computing (MEC) was first introduced by the European Telecommunications Standards Institute (ETSI) in 2014 as, "it provides information technology (IT) and cloud computing capabilities at the edge of the mobile network, within the Radio Access Network (RAN) in close proximity to mobile subscribers" [85]. MEC aims to deploy storage and computational services closer to the end users. It also enables third party applications and services at the edge of mobile networks. Recently, ETSI renamed its associated Industry Specification Group (ISG) as Multi-access Edge Computing (MEC).

Other closely related concept to MEC include cloudlets [86], [87] and fog computing [88]. However, typically cloudlet is referred to as the architecture in which the computational servers are located closed to the user premise (not at the RAN). In fog computing, the computational capability is integrated inside the IoT gateway which connects IoT devices. The concept of edge, cloudlet and fog computing are overlapping and these terms are frequently used interchangeably [21]. However, in the context of this paper, we will be referring to the MEC architecture in which the edge servers are located at the RAN, unless stated otherwise. Figure 4 illustrates the architecture of mobile edge computing. MEC allows storage and computational capabilities at the RAN level to provide several enhanced services to the end users.

MEC offers several potential advantages in live video streaming services.

- *Efficiency:* Using computational offloading, user devices can offload high computational tasks to edge servers for remote processing.
- *Ultra low latency:* By storing content closer to the users, delay associated with fetching content can be significantly reduced.
- *Available computation:* MEC can be used to augment capabilities of other devices, thus reducing the cost of transport.

Figure 5 illustrates the various benefits of MEC for end users as well as network operators in details. As depicted in the figure, these benefits are generally categorized into two categories: benefits to network providers and benefits to end users. Network providers can improve services reliability. As edge servers are distributed at RAN level, thus if one edge server is down or congested, users connected to other edge

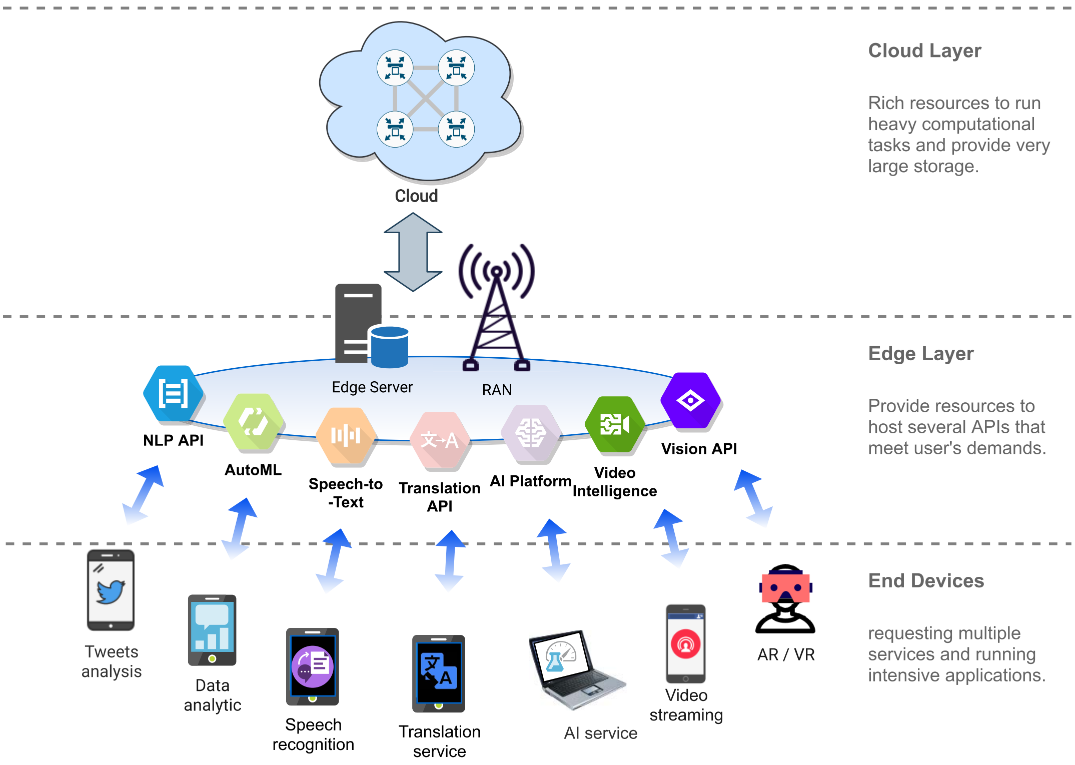

FIGURE 4. Mobile Edge Computing (MEC) architecture: storage and computational servers deployed at the RAN that enables a range of services to the network users.

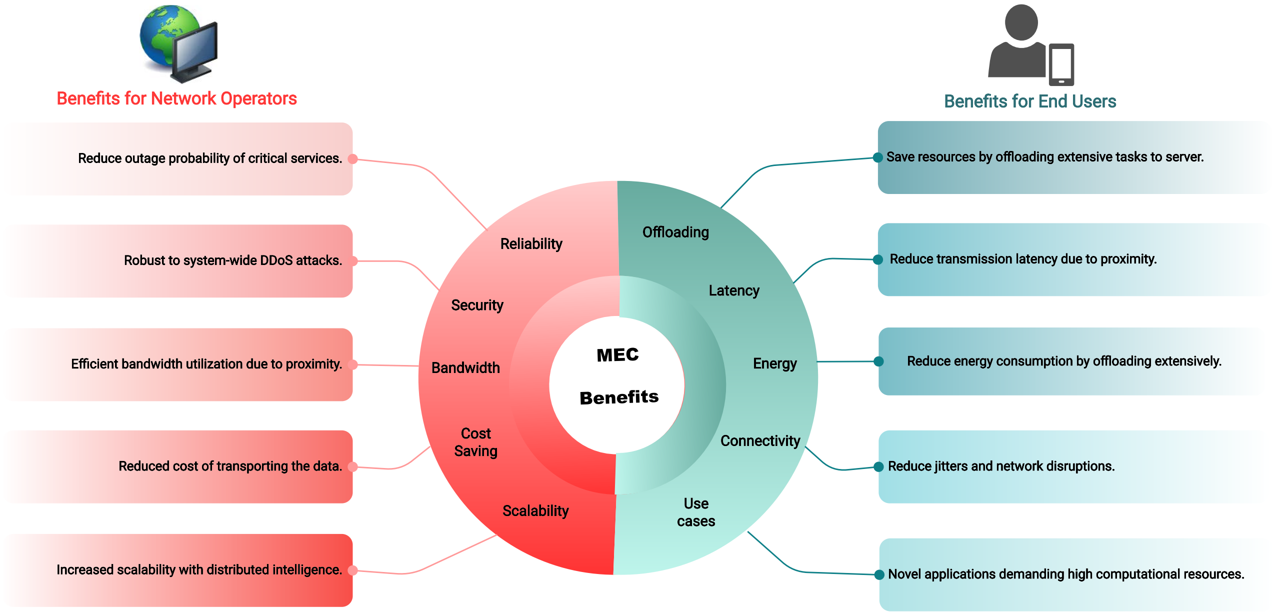
Benefits for Network Operators
Benefits for End Users
Reduce outage probability of critical services.
Robust to system-wide DDoS attacks.
Efficient bandwidth utilization due to proximity.
Reduced cost of transporting the data.
Increased scalability with distributed intelligence.
Save resources by offloading extensive tasks to server.
Reduce transmission latency due to proximity.
Reduce energy consumption by offloading extensively.
Reduce jitters and network disruptions.
Novel applications demanding high computational resources.
Reliability
Security
Bandwidth
Cost Saving
Scalability
Offloading
Latency
Energy
Connectivity
Use cases
MEC
Benefits

FIGURE 5. Benefits of mobile edge computing for network operators and end users.

servers are not affected. This is in contrast to cloud computing which has computing resources at one or fewer locations. Similarly, denial of service (DOS) attacks on MEC servers can affect users only the users at the effected servers, making edge-based applications more robust to DOS attacks. MEC also helps to alleviate network congestion at the backhaul links, by caching and processing locally at the edge servers whereas, in MCC the data has to be fetched from remote cloud servers. When the data is stored locally, more economical content can be provided to the users, providing more business opportunities to network and content providers. As compared to cloud-based centralized services, edge-based services are more reliable and secure, hence scalability comes as an inherited benefit provided by MEC.

In addition to the aforementioned benefits to network providers, MEC also bring a range of benefits to the end users. Users can offload their computation-intensive tasks to the edge server. By offloading computation and fetching locally cached content, the end-to-end latency experienced by end users can be significantly reduced. As mobile users are battery powered devices, mobile users can also exploit edge-based processing to save their energy consumption. When content is cached locally at the RAN (i.e. available at lower propagation distance or even at single-hop), video packets can be delivered with minimum delay and relatively less variations in packet delays, thus improving connectivity and reducing jitters. With the power of edge computing, mobile users can be enabled to run novel applications such as computationally intensive AI applications.

Mobile Edge Computing offers two major services i.e., providing (i) storage facility and (ii) computation resources, in the user proximity primarily to reduce latency with a range of associated benefits as illustrated in Figure 5. Using MEC, users can access locally stored content with low latency, and offload their computational tasks to the edge server for fast processing. In the following, the two concepts are explained in details. Furthermore, a brief review of recent research works in these areas is presented. (Note: The review of the state-of-the-art of MEC in video streaming services is presented in Section V).

B. EDGE-BASED CACHING

Traditional cloud-based architectures store millions of videos in relatively large-sized servers in few geographical locations. Edge caching is the process of fetching content from these remote cloud servers and storing at the local edge servers. Edge caching reduces data traffic transported over backhaul links and reduces the content delivery time. Edge caching techniques can be categorized in different ways e.g. proactive versus reactive caching, independent versus collaborative caching.

Proactive caching refers to fetching content before arrival of the request usually based on the video popularity or probability of the request, whereas reactive caching is post-request content fetching. Independent caching refers to the capacity of an edge server to solely decide content fetching based on local or central information, whereas in collaborative caching (also referred to as coordinated caching), multiple edge servers can provide their cached content to each other.

Any caching algorithm has two parts, i.e., *content fetching* and *cache replacement*. Content fetching refers to the process of bringing content from the remote server (origin or cloud storage) and storing it into the local/edge server. Cache replacement refers to the process of selecting content that need to be removed from the local/edge server if there is limited storage size available to store the newly fetched content. Content caching in traditional networks such as content centric networks have been extensively investigated in several studies [89], [90]. In edge computing, the caching problem is typically studied as a joint problem with the offloading decision [91], [92]. A detailed review of caching schemes in video streaming context is provided in Section IV.

C. EDGE-BASED COMPUTATION

Computation at edge nodes can be implemented in several ways depending on the design objective. Common design objectives for implementing edge-based computation and optimization include energy consumption, computation latency or bandwidth utilization. In this section, we present a brief overview of different computation offloading models in mobile edge computing and a summary of some representative works of each category. As illustrated in Figure 6, the computation models are categorized into three classes. A similar classification of computation offloading in MEC networks can be found in [21].

1) SINGLE USER SYSTEMS

This is the simplest computation offloading model that consists of a single edge server and a single user. Although real networks always have multiple users, the purpose of the model is to simplify the computation offloading decision by considering the task of every single user independently. The model decides whether a particular task of a user should be computed locally at the device or offloaded to the edge server. The model can be sub-classified as (i) *binary offloading* and (ii) *partial offloading*.

If the whole task is computed locally or offloaded wholly to the edge, this is referred as "binary offloading". In contrast, when a user device computes a part of the task locally whereas offload the remaining part to the edge, the model is referred as partial offloading. Examples of binary offloading include [93], [94], [95]. In [93], authors investigate the simple case of single user to analyze computation offloading while minimizing the energy consumption at the mobile device under a computational rate constraint. The model considers various parameters including device's and server computational power and the communication bandwidth to take the offloading decision. A task is offloaded to server if the energy consumption by the user device for offloading the task is less than the energy needed for local computation. The energy consumption is studied as a function of distance between the device and edge server. The results show improved energy performance at shorter distance which increases when the distance increases. The delay in this work is considered as a hard constraint. In [94], authors propose an alternate method to solve the binary offloading decision to run mobile application locally or at cloud server. The objective is similar to [93] i.e., to minimize the device's energy consumption, while considering soft delay deadlines. The energy consumption in this work is modeled as a function of CPU frequency:

$$\xi(f) = kf^2 \tag{1}$$

where, k is the switched capacitance (set to 10^{-11}) depending on the processor chip architecture. To minimize the energy in local computing mode, the CPU frequency is adjusted, whereas in remote computing mode, the data transmission rate is adjusted. A limitation of this work is that in local mode, the energy consumption is reduced by reducing CPU cycle frequency which means the application will run slowly, which make it unsuitable for heavily intensive applications with delay constraint. In [95], binary task offloading is proposed to maximize the revenue of the service provider. This paper does not focus on the QoS requirement of the user/application. Other works on binary offloading can be found in [96] and [97].

Partial offloading schemes in MEC systems have been investigated in [98], [99], and [100]. In [98], authors proposed a partial task offloading model in which a single task is divided into sub-tasks and then each sub-tasks is either computed locally or offloaded to the edge. The task offloading depends upon the local computation resource available, the communication channel capacity and the queue size in the edge server. The offloading decision is optimized using a greedy algorithm (called as Select Maximum Saved Energy First (SMSEF) algorithm) to maximize the energy saving at the mobile devices. However, this scheme does not minimize the task completion delay, but only ensures that the delay deadline is met. In [99], authors proposed a partial computation offloading scheme by jointly optimizing the computational speed, transmit power and the offloading ratio. The study aims at minimizing both the energy consumption and the computation time. The energy consumption is modeled similar to Eq. 1. In [100], authors proposed a heuristic that makes an online offloading decision on sequential tasks. The algorithm aims to minimize the completion time of the application. The task completion time for local and remote execution is modeled as:

$$\text{Execution time} = \begin{cases} \frac{s}{\mu_m} & \text{Local} \\ \frac{s}{\mu_c} + \sum \frac{s_i}{\mu_c} + \frac{d_{i,j}}{R} & \text{Remote} \end{cases}$$

where, s_i is the task size (as number of instruction), μ_m and μ_c are the CPU capacity of mobile device and the edge server respectively, and R is the data rate of the wireless channel. Other works on partial offloading can be found in [101], [102], [103], and [104].

Binary offloading is preferred for problems involving similar tasks and in which user device has strong channel conditions to the base station for fast computation. The benefits are to reduce the overall latency reduction of the system and energy saving at the user device. On the other hand, partial offloading is a preferred model for problems involving heterogeneous tasks, thus offloading computational intensive portion of the task to the edge whereas perform the remaining tasks locally. Similarly users can decide to offload tasks of small transmission payload to the edge to reduce data loss.

The aforementioned two single user MEC models are used when the task arrival process is deterministic. However, in some cases, the task arrival is random i.e., tasks arrive at the processor queue at a random rate. In such a case, the offloading decision is also dynamic and is referred to as *stochastic offloading*. Stochastic offloading becomes a necessary requirement in some problems e.g. if the random task arrival rate is leading to buffer overflow, dynamic offloading can help to avoid this by randomly offloading tasks to the edge when the task arrival rate increases. Another use case is to exploit the randomness of the wireless channel to design channel-aware offloading schemes in which tasks are offloaded only when the channel supports the required communication delay. Example works on stochastic offloading can be found in [105] and [106]. As compared to deterministic task arrival rates, stochastic tasks arrival requires more robust techniques to optimize the offloading decisions. Authors in [105] use Q-learning to perform the offloading such that the tasks are completed within the delay deadlines, thus achieving the end-to-end reliability. Similarly, authors in [106] studied the resource allocation problem in dynamic scenarios. The objective is to reduce the unnecessary resource allocation, maximize the QoE and minimize the network cost to the content provider in a multi-edge servers environment. The authors used several machine learning (ML) algorithms (Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Network (CNN), MultiLayer Perceptron (MLP) and XGboost) on synthetic datasets to evaluate the effectiveness of each model. The use of ML based techniques achieves better performance due to their capability to accurately estimate the stochastic parameters in real-world applications.

2) MULTI USER SYSTEMS

Multi users system refers to the computation model in which multiple user devices share a single edge server. In such systems, computation offloading can be implemented in various ways. For instance, the tasks offloaded by multiple users can have different priorities, hence the server must schedule the tasks computation according to the priorities of the tasks. Similarly, different users have usually different channel conditions and consequently support different data rates, hence a "joint resource allocation" model which jointly optimizes the offloading and radio resources is desired to improve system-wide performance. Examples of joint optimization of radio and offloading in multi users MEC systems are studied in [107] and [108]. In such kind of problem, one of the parameter is taken as a hard constraint, while optimizing over the others. For instance, in [107] the mobile energy consumption is minimized under average latency constraint, whereas in [108] the weighted-sum of latency is minimized while considering/satisfying the mobile devices' energy consumption requirements. The joint optimization over two parameters does not preclude to consider other parameters such as computational and storage resources.

As the edge server also provides caching facility to store content, the joint optimization of caching and processing can provide several benefits such as service provider revenue or content retrieval latency and storage utilization. Joint caching and processing systems are studied in [91] and [92]. In joint caching and processing, the objective is to find the optimal location (i.e., edge/cloud servers) for storing and processing the content. For instance in [91], authors consider jointly the storage and processing resources to minimize the backhaul network cost of serving all requests. This is done by jointly determining the cache placement and processing scheduling using Integer Linear Programming (ILP).

In multi user systems, when users offload tasks to the edge server, the tasks may have different priorities. To meet the QoS of such applications, the edge server may schedule these tasks based on the latency requirements. Hence, latency sensitive tasks are computed first, followed by latency-tolerant tasks. Examples of server scheduling include [109], [110], [111]. Authors in [109] proposed task scheduling based on the energy saving opportunity. Thus, a device offloads the task to the server when the offloading can achieve more energy saving than local computing. However, the offloading method in this work is binary and task splitting was not investigated. In [110], authors propose the task scheduling under strict delay requirement. The work proposes to use dynamic voltage and frequency scaling (DVFS) technique for local computing whereas adjusting transmission power for remote computing (edge/cloud) to reduce energy consumption in both modes. In [111], authors further investigated the scheduling problem by jointly considering server scheduling and video bitrate selection to improve the overall QoE and fairness in resources allocation.

Unlike the cloud computing paradigm, in edge computing, the edge servers have limited computation power. Furthermore, serving heterogeneous users requests becomes more challenging when the server has limited resources. To overcome the resource limitations, recently, MEC operations have been investigated with device-to-device cooperation. D2D-assisted MEC systems have been recently proposed in [112], [113], [114], and [115]. In D2D-assisted MEC systems improve system's overall capacity in two ways. First, users can exploit D2D cooperation to offload tasks to its neighboring devices, instead of offloading to the resource limited server. Secondly, MEC servers can itself offload computation tasks to user devices.

3) MULTI SERVER SYSTEMS

Multi Server systems refer to the MEC systems consisting of multiple and typically heterogeneous MEC servers. Multi server systems offer several challenges. The first challenge in multi-server system is the selection of server for offloading computation. Different selection criteria can be considered e.g. selecting closest server, selecting least loaded server, selecting between cloud or edge server. For example, in [116], authors proposed a scheme to select the closest server to minimize the resource allocation cost and maximize QoE. The work used the FaceBook 2018 live video dataset to predict the viewing patterns and automatically select the edge server to minimize the startup delay. The work employs ML techniques such as Multilayer-perceptron (MLP), Decision trees (DT) and Random Forest (RF). Similarly, in [117], authors propose a load balancing and task allocation scheme using Particle Swarm Optimization (PSO). The users select the edge server based on multiple criteria i.e., current server load and distance to the edge server.

Multi-server systems can benefit from the cooperation among the servers. Servers can collaborate with each other in several ways. For example, servers in a small geographical area connected by single hop cooperate by offloading computation to each other. Similarly, edge servers can collaborate with cloud servers, to compute delay-sensitive tasks locally (at the edge server), whereas offload delay-tolerant tasks to the cloud servers. Examples of edge-cloud cooperation are [108], [118], and [119]. In [108], cloud-edge cooperation is used to split tasks among edge and cloud to minimize the latency. On the other hand, [118] proposes to schedule tasks such that delay-sensitive tasks are processed at the nearest resource-constraint edge server whereas delay-tolerant tasks are offloaded to the remote resource abundant cloud servers. Similarly, in [119], authors propose the cooperation among multiple edge servers to share contents via backhaul links when requested.

In multi server systems, a mobile user may move away from one server and get closer to another MEC server. Consequently, the network controller can optimize computation in two ways: (i) by offloading computation to the new server or (ii) performing computation on the origin server and then forwarding the computation results to the new server. Example works on computation migration are available in [120], [121], and [122]. In [120], authors used Markov Decision Problem (MDP) to formulate the computation migration in multi-server environment. The migration decision is taken considering the distance between the users to each server and using two thresholds. The work is extended in [121] by jointly considering the computation scheduling and computation migration to minimize the average transmission energy and reconfiguration cost. In [122], authors proposed to compute the tasks locally or migrate computation to remote cloud server such that the total energy consumption and latency is minimized.

Table 4 provides a brief summary of representative works on resource allocation schemes in edge computing. In Section V, we provided a details review of the state-of-the-art on resource management in video streaming applications. A more exhaustive list of research contributions on resources management in MEC in generic applications is provided in [15] and [123].

D. LESSONS LEARNED AND CHALLENGES

Edge computing provides storage and computation resources closer to the mobile devices i.e., at the base station. In the caching process, content is pre-fetched and stored at the edge server. As edge storage capacity is limited, older content need to be replaced when it is reaching the available storage capacity. Content caching can be implemented in

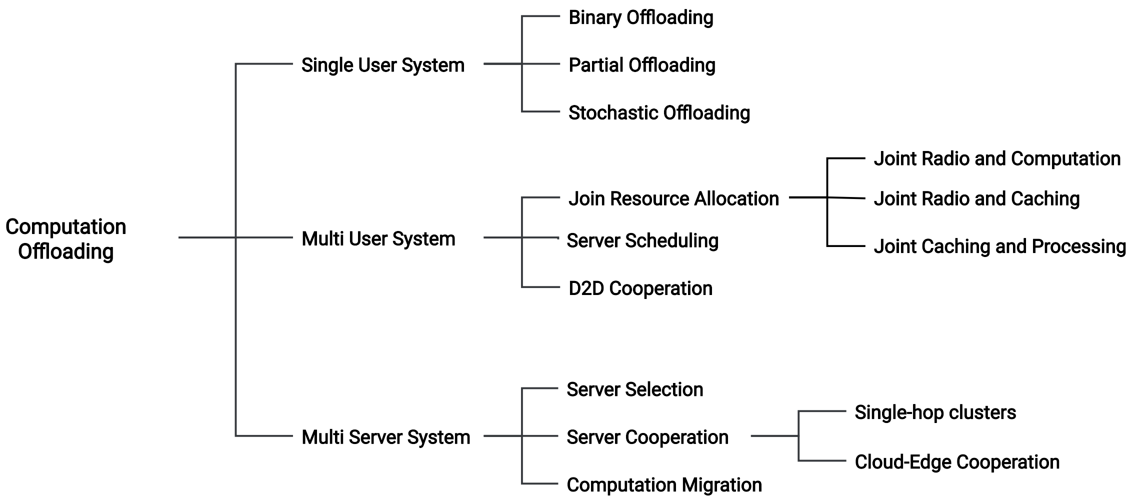

FIGURE 6. Computational offloading in mobile edge computing.

non-cooperative method such as in [124] and [125] or cooperative method [126], [127]. Cooperative caching methods are more popular due to their system-wide performance gains. The computation power of edge server is also limited and hence efficient resource allocation strategies are required. In the simplest case (single user system), a mobile device connected to an edge server can offload computationally intensive tasks to the edge whereas perform simpler tasks locally. Alternatively, mobile devices can also offload sub-tasks to the edge (i.e., partial offloading). In the case of multi user systems, edge server can schedule offloaded tasks according to the latency requirements of the user's applications.

Due to the high number of decision factors and the system uncertainties, the resource allocation problem in mobile edge networks becomes highly complex. More specifically, to optimize the MEC resource utilization, multiple decisions should be considered, including the caching, computing and networking variables. The main metric related to the caching is the hit ratio, whereas the metrics related to the computing are the latency, the throughput, and the energy consumption. On the other hand, the transmission latency, the data rate and the QoE, should be taken into account while scheduling transmission decisions. Resource allocation problems are typically addressed by formulating an optimization problem with one or multiple objective functions using different system constraints. The traditional techniques to solve such problems are included mainly under the umbrella of stochastic and convex optimizations and game theory. However, these approaches are very complex and time consuming and are not adequate for online implementation. Recently, reinforcement learning gained a lot of attention owing to its ability to solve resource allocation systems, particularly, those with dynamic, large and complex problem spaces. RL-based approaches are still in their infancy and further efforts need to be conducted to examine their performance on MEC networks.

Further improvements can be achieved by considering the users radio channel in the joint optimization scheme. The joint optimization of edge resources can improve the resource utilization to a certain level, however recently, D2D collaboration has been proposed in edge computing system, which enabled mobile devices to offload computations to other resource-rich helper devices. Such types of D2D-enabled MEC systems can boost the performance of the system beyond the edge capacity limits.

IV. MOBILE EDGE COMPUTING - APPLICATIONS IN VIDEO STREAMING

In the previous section, we discussed the detail overview of mobile edge computing, its architecture and state-of-the-art including caching and computation. However, it is important to understand how MEC can help to improve the users' video streaming experience and/or implement novel video streaming applications and services. The primary benefit of mobile edge computing in the video streaming is reducing the end to end latency to enhance user quality of experience by eliminating network lags, frames dropping and buffering. However, in addition to latency reduction, MEC alleviate the network congestion by running applications and performing the processing tasks closer to the end users. As the MEC servers are deployed at the base station (i.e. RAN), thus it allows flexible and rapid deployment of new applications and services for cellular customers. Similarly, as network service providers can authorize trusted third parties, such as application developers and content providers to deploy MEC-based services.

This section outlines novel video streaming services in which MEC can bring potential advantages. It is evident from the previous discussion in Section III that MEC helps improve video streaming via optimal caching of contents at the network edge and provide computing resources for faster processing. However, how these inherited advantages of MEC benefit video streaming? What are the scenarios

TABLE 4. Computational offloading schemes in mobile edge computing.

Category	Ref	Description	Evaluation Metrics	Method/Algorithm
Binary Offloading	[93]	Transmission energy minimization under computation deadline.	Delay, Energy	Convex programming
	[94]	Minimize energy consumption with a soft real-time requirement.	Delay, Energy	Convex Programming
	[95]	A binary offloading model to maximize profit to video service provider.	N/A	MAB
Partial Offloading	[98]	Schedule offloading tasks from multiple mobile nodes to a single MEC server in order to maximize energy savings of all mobile nodes.	Energy	SMSEF
	[99]	Task-input data is divided for local and remote execution.	Energy, Latency	uni-variate search
	[100]	Load balancing between mobile and edge servers to minimize latency.	Latency	Heuristic
Stochastic Offloading	[105]	Reinforcement learning to proactive allocation resources to multiple.	E2E Reliability	Q-learning
	[106]	Transcoding resource allocation using past video requests and then using time series forecasting to predict the resources needed for future requests using a heuristic.	QoE, Network cost	LSTM, GRU, CNN, MLP, XGboost
Server Scheduling	[111]	A scheduling algorithm that solves client to edge mapping (load balancing) and per client bit rate selection problems.	Throughput, Buffer delay, Fairness	Heuristic
	[109]	Scheduling uplink and downlink transmissions using queuing theory.	Energy	Nested interval algorithm
	[110]	Tasks Offloading selection using clock frequency and transmission power allocation.	Energy, Delay	DVFS
Joint Radio and Processing	[108]	Jointly optimize communication and computational resources with tasks splitting between edge and cloud.	Latency	Convex optimization
	[107]	Joint allocation of computation and radio resources.	Power consumption	Convex optimization
Joint Caching and Processing	[91]	Jointly optimize caching and transcoding resources to minimize the backhaul network cost.	Network cost	ILP
	[92]	Collaborative offloading and caching scheme using Lyapunov optimization to minimize the overall latency of all mobile devices.	Throughput, Playout delay, Buffer length, Rebuffering duration	Theoretical architecture
Server Selection	[117]	Balance work load among edge nodes by considering edge computing capability, existing work load and distance among edge server and user.	Service delay	PSO
	[116]	Proactively allocate resources by predicting the number of viewers in a cloud site and then serving viewers by their closest server.	Viewers' QoE, Network cost	MLP, DT, RF
Server Cooperation	[119]	edge servers in a sing-hop clusters cooperate to cache and transcode contents.	Network cost, delay, cache hit ratio	ILP
	[108]	Jointly optimize communication and computational resources with tasks splitting between edge and cloud.	Delay	Convex optimization
	[118]	Edge/cloud selection to meet deadline requirements.	Task completion	Heuristic
Computation Migration	[120]	Computation migration based on distance threshold between user and two servers using MDP.	MDP	Delay
	[121]	Jointly optimize computation scheduling and service migration to minimize energy transmission and reconfiguration cost.	MDP	Application queue length
	[122]	Compute locally or offload to remote server such that total energy consumption and latency is minimized.	Energy, Delay	MIP

and novel services that can be realised using MEC? This section provides an overview of the attractive applications of MEC-based video streaming services. Figure 7 illustrates the most attractive applications of MEC assisted video streaming, whereas specific examples of each application category are listed in Table 5.

A. CONTENT SEARCHING

The powerful computing capabilities of the edge server can bring a more personalized search features which runs faster than traditional approaches using cloud computing. Using MEC, locally available information at the edge servers about user preferences, location, local events and incidents can help improve user search experience [17], [128]. For instance, users in a specific area when search for a local event or incidents, the content searching can be optimized based on the other local users' search history and selections data saved at the local MEC server. This kind of localised content searching is only possible using MEC servers deployed in users' proximity.

B. CONTENT SUGGESTION

MEC offers the computational capabilities to process the data related to user preferences and activities. When such processing is done faster, the user can be served with appealing content in real time that ultimately enhancing user streaming experience [128]. The kind of location specific content suggestions is based on the search history of other local users, which is saved in the edge caches.

C. TARGETED ADVERTISEMENT

When powerful computations are applied to user's preference data, the usually unwanted advertisements that interrupt streaming videos can be tailored to the user's interest.

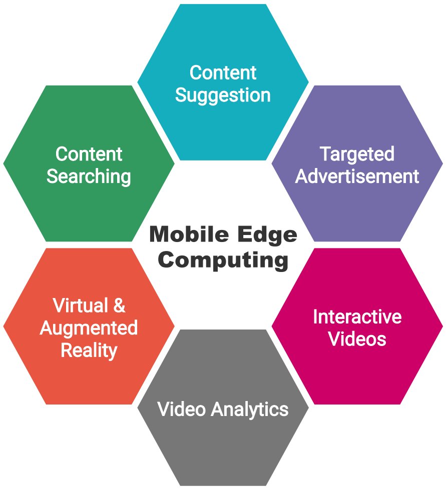

FIGURE 7. MEC use cases in video streaming applications.

TABLE 5. MEC applications in video streaming.

Content Searching	
Faster search	Faster content search with user's search history cached and updated over time.
Personalised search	Personalized search based on user's location (e.g. supermarket, office, stadium)

Content Suggestions	
Use case 1	Personalized content suggestion based on user preferences.
Use case 2	Improved content suggestions over time when user preferences changes.
Use case 3	Content suggestions based on local/regional events.

Targeted Advertisements	
Use case 1	Highly targeted advertisements based on user's interests.
Use case 2	Targeted advertisements based on user's previously visited shopping locations with timing information.

Interactive Videos	
Live Sports	Display Live statistics of a match
Filmography	Pulling filmography of an actor appearing in a scene
Retail/E-commerce	Allowing users to click on objects inside video to see details or buy.
Education/Trainings	Allow users to ask questions, submit answers, fill surveys

Video Analytics	
Surveillance	object detection, motion tracking, facial recognition, gesture recognition , activity recognition, head counting
Content detection	detecting banned videos, parental control, illegal videos
QoE Measurement	KPI calculation for encrypted streaming videos
Realtime Assistance	Detecting obstacles using videos captured by stereoscopic camera

VR and AR	
Retail	Try items using AR/VR reality without physically trying things
Gaming	Pokémon GO is a popular example.
Education/Trainings	Wearable cognitive assistance
Tourism	Try items using AR/VR reality without physical try things
Healthcare	AR enabled minimal invasive surgeries
Industry/safety	Building Digital maps and digital twin

Advertisement agencies can benefit from the user search preferences, search history, previous purchase history, locations visited with timing information etc., to send highly targeted advertisements [129]. As an example, a user at a shopping mall would be probably buying some goods whereas another user at a hospital would be interested in a healthcare product. MEC thus allows location-based service recommendations tightly coupled with a specific place. Particularly, information of radio node to which user terminals is connected is available at MEC server that can be used to get location information of users especially when users are connected to an indoor small cell. Then an inference engine deployed at the MEC server can determine proper services and sends the related advertisements for the user at the moment.

D. INTERACTIVE VIDEO EXPERIENCE

Users would always enjoy interacting with live videos e.g. displaying the statistics of a baseball game being streamed, pulling the filmography of an actor as he appears in a scene. Such user interaction always requires a level of immediacy which can be made possible using the high computation power of the edge servers available with minimum latency [130].

E. VIDEO ANALYTICS

Surveillance and video analytics to detect accidents are included under the umbrella of live video streaming. More specifically, in surveillance applications, the aim is to monitor a specific area and identify potential threats within the target region. Some of the area are very critical and need 24/7 surveillance such as military borders, oil/gas off-shores, and forests exposed to potential fires. The video frames (or video streaming) are sent instantaneously to remote servers for real-time object/accident detection using AI for example. The traditional wisdom resorts to cloud or servers to compute these heavy tasks. However, video streaming destined for detection and surveillance do not tolerate high latency, such as forest fire detection that needs immediate intervention. Furthermore, used cameras are sending high-resolution video frames to cloud servers and knowing that incidents are rarely occurring, the large data volume transmitted by source units has become problematic, particularly for systems that do not have stable bandwidth availability. Because of this tremendous amount of data, video analytics should be done at the edge of the network.

The huge amount of videos generated by mobile users, social media, IoT devices, scientific apparatus, satellites, and video surveillance cameras is being processed by modern computer vision techniques powered with AI. The processing can be done on-camera, which requires expensive AI-powered chips. Alternatively, to process these videos on cloud, longer delays occur as previously described (approximately 150 to 200 milliseconds). Edge computing offers a good trade-off by eliminating the need for on-camera processing to reduce cost and processing videos locally reduces delay (around 10 milliseconds). Reduced delay allows for quick detection and faster response which is required in many applications.

Video streaming analytic has a broad range of applications such as head counting in live streaming videos, suspicious activity detection, correlations in different video streams, combining real time information with historical context and search and rescue in live videos captured using drones. All these use cases require low latency (sometimes ultra low latency) analysis of streaming videos, thus motivates for using edge computing. Representative works in video analytics can be found in [131], [132], [133], [134], [135], [136], and [137].

F. VIRTUAL REALITY (VR) AND AUGMENTED REALITY (AR)

VR and AR are two emerging technologies that connect the physical and digital worlds. While there is a distinction between both,[1] both uses streaming videos to enrich user experience. There are a range of attractive use cases of both technologies such as games, entertainment, training, education and scientific areas [138]. A list of games using VR/AR technologies can be found in [139].

Recently many commercialized smart applications [140] are proposing to create a virtual avatar of the user, that allows to try the clothes virtually. More specifically, the user is able see his/her reflection on the screen as if he/she is looking to a mirror, where the virtual clothes are blended with the scene. This process is called blended-reality. The application users can see themselves moving, turning around, and walking while being dressed with the chosen items. The users can also see, in 360°, how the item looks like on them and if the size is appropriate. In the same context, Amazon has published a patent [141] of its partially reflective intelligent mirror, where virtual clothes and real scenes are transmitted through the mirror to generate a blended-reality client able to see himself/herself wearing new items. This type of application, called interactive application, uses virtual reality streaming that is extremely intolerant to delay variance or image freezing, as the user wants to see his/her avatar without any motion sickness affecting the quality of experience. For this reasons, the cloud wisdom to blend the streamed video is no longer sustainable of such real-time applications and sending data to remote severs may not satisfy the latency requirements.

MEC can be used to improve VR/AR applications for several benefits such as latency reduction [142], [143], efficient resource utilization using device-MEC collaboration [142], improved throughput [142], and reducing the backhaul traffic load [144].

G. LESSONS LEARNED AND CHALLENGES

Mobile edge computing can help improve video streaming services by providing proximal caching and computational resources for transcoding the videos. In addition, faster edge-based processing of videos, a number of benefits can be achieved such as faster content searching and personalized suggestions, targeted advertisements, real-time user interaction with videos, video analytics for surveillance, object detection and real-time assistance, and virtual reality applications such as gaming, retail, healthcare and industrial safety. It is widely believed that 5G networks fall short of meeting the unprecedented requirements of data-intensive and delay-sensitive applications such as autonomous vehicles, remote surgeries etc., thus leaving room for acceptance of edge-based solutions. The future 6G networks further mandates the use of edge intelligence in a variety of network functions and services. However, the success of MEC-based solution heavily rely upon the edge infrastructure deployment. MEC deployment will incrementally progress in a way to support as new network services are introduced. As will be explored further in the subsequent section, MEC deployment considerations include revenue as an important factor in addition to the QoS metrics (e.g., delay, storage, throughput, energy etc.).

V. STATE-OF-THE-ART IN MEC-BASED VIDEO STREAMING

In the previous section, we discussed MEC-assisted video streaming applications and services. However, to implement these services in real-world applications, providing caching and computing resources along the network edge is not sufficient. Indeed, it involves various challenges such as efficient caching strategies, optimal resource allocation, cooperation among network entities, and tasks/requests scheduling, etc, to realise the full range of these applications. There have been a huge amount of research efforts contributed to cope with these challenges. This section focuses on covering these research works to solve the aforementioned problems. We have dedicated separate sections for the works involving device-to-device (D2D) cooperation (in Section VI) and machine learning (in Section VII).

[1] AR overlay digital elements such as visual content and information on your real world view, whereas VR implies a complete immersion experience, allowing users to experience things and places that actually do not exist there in the user's environment. Mixed Reality (MR) refers to combining VR and AR technologies for a richer user experience.

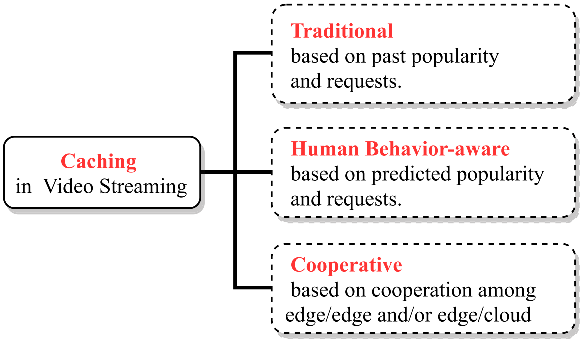

FIGURE 8. Caching techniques in video streaming.

A. CONTENT CACHING AT EDGE

As discussed earlier, MEC offers proximal storage at the edge server that significantly reduces the delay to retrieve content and alleviate the congestion and bandwidth usage for network operators, resulting in overall improved performance and QoS in several services [131]. However an edge server has limited storage as compared to cloud servers to store large-sized video content, hence caching schemes need to be optimized for efficient allocation of storage resources to end users. In the following, we present research contributions aiming at edge caching optimization.

One of the key parameters that influences the content retrieval latency and the traffic transmitted over the network is the distance between the end-user and the remote servers. In this context, caching the video content in MEC servers is a promising solution that enhances the bandwidth efficiency and reduces the latency to serve viewers. However, the edge servers are characterized by their limited capacities in terms of storage. On the other hand, we are witnessing an explosion of newly published video streaming. For example, in 2020, users have been uploading to Youtube around 500 hours of videos every minute [145]. Therefore, edge caching should be performed wisely. Several approaches responsible for scheduling the content caching and removal have been proposed in the literature. These approaches can be classified into two groups, namely traditional schemes and popularity/human behavior- aware strategies.

1) TRADITIONAL CACHING SCHEMES

The traditional schemes assume that each video has a popularity assessed by the number of views and this popularity remains stable, even after passing the trend. Several policies are proposed in this context:

- *Most Popular Videos (MPV):* This technique was originally used by Hulu [146] for content caching. MPV is a proactive caching policy that caches the content based on the nation-wide content popularity. The shortcoming of MPV is that the cache is not updated based on the user requests or viewing patterns. MPV is suitable for large caches such as Internet CDN to achieve a high hit ratio but performs poorly on edge caching due to the fact that the local request by users may be different than the nation-wide distribution.
- *Least Recently Used (LRU):* This conventional caching scheme can be considered as a baseline that has been used for a long time in networking systems. LRU stores the time of the last access of each content and when the memory capacity of the server is insufficient, it replaces the most idle video that was not requested for a long time, by a new content. The problem of LRU is that it gives priority to some unpopular content just because they were recently requested [147]. LRU has multiple variants including the three Segmented Least Recently (S3-LRU), which divides the cache into three segments where the most requested videos are stored in the highest segments and least requested are saved in the last segment. If a new content is requested, it is placed in the head of the list while removing the tail of the last segment. If it already exists in the cache, it is considered as the least recently requested and all the others are shifted downwards [148].
- *Least Frequently Used (LFU):* This caching strategy relies on the number of requests per video to judge the popularity of content. In this way, the content with the lowest number of requests is evicted to create a room for the new published video. However, LFU suffers from gradual performance degradation as videos with historical high number of requests can remain a long time in the cache even if they are no longer accessed [149]. This problem is solved by defining a time window for observing the frequency of requests as done in the Jumping Window Least Frequently Used (JW-LFU) and the Sliding Window Least Frequently Used (SW-LFU).
- *First In First Out (FIFO):* This scheme caches the videos according to the order of their first request. It means when a new video is requested for the first time, it occupies the head of the list. The FIFO was later combined with LFU (FIFO-LFU) to take into consideration the popularity of the content [150].
- *Least Recently Frequency Used (LRFU):* This caching approach takes into consideration both frequency of requests and recentness, as adopted in [151].

The aforementioned traditional caching schemes are widely used in edge computing owing to their low complexity and simplicity of deployment. However, these approaches ignored the dynamic of viewers, and their preferences and behaviors. Therefore, due to the limitation of edge bandwidth and memory, better caching schemes that take into consideration the viewership and their capacities and make videos available at peak hours beforehand using prediction techniques should be designed.

2) HUMAN BEHAVIOR-AWARE STRATEGIES

This type of strategies is based on popularity prediction and the learning of the viewers' preferences through either AI

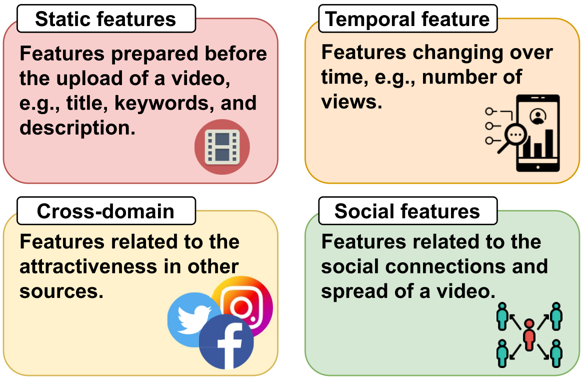
Static features
Features prepared before the upload of a video, e.g., title, keywords, and description.
Temporal feature
Features changing over time, e.g., number of views.
Cross-domain
Features related to the attractiveness in other sources.
Social features
Features related to the social connections and spread of a video.

FIGURE 9. Popularity prediction features.

or mathematical studies. However, prediction techniques can add latency overheads and computational costs to ensure satisfactory performance and take accurate caching decisions. Therefore, the designed algorithm should be quick, provide accurate decisions, and scalable to handle high number of requests and large library of content. Finally, the prediction should be based on the preferences and behaviors of in-proximity viewers. Specifically, the contextual and social information, the content' demands, the users' interests, and the geographical location of viewers should be the input of the prediction model. These data is processed to get spatial, temporal and social insights and identify the viewing pattern. In this way, the designed caching strategy will be able to predict the videos that will receive higher attention locally or globally, and accordingly plan for content storage, replication, eviction and select the most adequate MEC server for content storage. The input features can be classified into four groups: static, temporal, cross-domain and social features as we can see in Figure 9.

- **Static features:** refer to the parameters that are prepared before publishing the content. These features include the video characteristics that give hints about the quality of the video such as the duration, publication date, video and audio standard, music style, etc. The visual features such as the images at each frame and the text features such as the category, the title, the keywords, and the description can contribute to increase the popularity of the video and enhance its visibility in the search engines.
- **Temporal features:** refer to the data that change over time such as the channel features including the number of rates, views, subscribers, comments and shares for different published videos and the video features such as the video age, the watch time, and the subscribers gained from this content. Additionally, the previous requests of viewers can give insights about their future requests for other content.
- **Cross-domain features:** refer to the external sources such as the reputation of the content creator and the propagation of the content via other social medias or video platforms.
- **Social features:** includes the relationship between users, their followers and followees, and their social interaction which can help to share and forward the content to more viewers.

A wise selection of the features fed as an input to the prediction model is very important to improve the accuracy, reduce the complexity and computation time of the decisions, and remove the redundancy of attributes. Next, we will discuss the popularity-aware MEC caching approaches that rely on the described features. These approaches can be classified into two groups: a single domain and cross-domain strategies. The single domain covers the approaches trusting only the features related to the video and its broadcasting platform such as:

- *Popularity evolution of a content:* The high correlation between the past popularity of a videos (e.g., number of views, shares, and requests) can indicate its trend for the future. In [125], two caching policies i.e., P-UPP and R-UPP are proposed, which are based on the popularity of content. In P-UPP (Proactive User Preference Profile) scheme, videos that are Most Likely to be Requested (MLR) by active users of the cell are pre-fetched. This is different from the traditional scheme such as MPV, which accounts for the nation-wide popularity of the video content. The local popularity of a video v_i in a pool of videos V is calculated using the equation:
$$p(v_{i,j}) = \frac{p_{vc_j}(v_i)}{\sum_{i=1}^{|V|}(v_i)} \tag{2}$$
where the denominator of the equation represents the probabilities of all videos in the category vc_j. A drawback of this scheme is that if the cell is highly dynamic, the users may leave or join frequently which will change the MLR set of videos, thus increasing the computation. To reduce the computation complexity caused by cell dynamics, authors proposed to use a threshold value for cache hit ratio to decide before pre-fetching/replacing content. In R-UPP (Reactive User Preference Profile), a video is fetched upon user request, however if the cache is full, the video that is ''Least Likely Requested'' (LLR) is replaced. LLR is calculated using the probability of requesting a video. In real-world, the demand of a video content is always associated with the user behavior i.e., it is the user preference that makes a video more popular. Hence, caching schemes based on the user context/behavior is proposed in [152]. The authors proposed a smart caching scheme using knapsack problem to find context-aware content popularity to maximise video hit rate within a limited time. As the user preference may change based on time and/or location, in [152], the spatio-temporal context is constructed from the user's access time and spatial characteristics to mimic video popularity. In [153], a collaborative online caching algorithm is proposed that minimizes the sum of User Attrition (UA) cost and caching cost for each content. The problem is formulated as Integer Linear Program (ILP).

In the online settings, the user requests are revealed one by one to the edge server and the server should take decision before next request to provide the content from local cache (if available), or fetch from neighboring edge servers or from the origin server. Authors in [154] proposed Proactive Cache Policy (PcP) i.e., a popularity-aware proactive video chunks caching based on chunk popularity instead of the whole video popularity. This is particularly useful as normally the users would not watch the whole video but rather watches the first few chunks. Hence, this work proposes to cache the popular chunks rather than the entire video of long duration.

- *Metadata of the content:* Some videos are newly published and has not indication about the past. Hence, the content data such as subscribers, title, and frames can be considered to predict the popularity. Authors in [148] proposed an online caching strategy that uses the mixed-integer linear program (MILP) to load the MEC cache with most popular videos during the valley hours, based on the content metadata such as keywords and description. Authors in [155] designed a deep learning approach that handles the popularity of new published videos. The input data of the model are extracted from the video raw data. Then, the popularity is defined by studying the similarity of the extracted features to the old videos.
- *Social dynamics:* the interaction between users (e.g., friendship, likes, and shares) are very insightful for the watching trend of videos within a small edge area. The authors in [156] used the susceptible-infected-recovery (SIR) model to study the social propagation of videos. More specifically, the SIR model defines three states: susceptible, informed and refractory. In the first state, the viewers are notified of the new published content. During the informed state, the users discover the video and decide whether to share it to the followers or not. If the content is shared, more viewers can see it, if not the initial user is considered in refractory state. The popularity of a video can be examined through the probability to spread the content, which is also called social connections.

The cross-domain prediction models use features from external sources (e.g., attractiveness of the video on other sources) to increase the accuracy of the decision. In [157], authors improve the popularity-aware video caching by jointly considering the popularity and attractiveness of the video stream. The authors argue that despite the high popularity of the videos, user may decide to finish the videos after a short duration of viewing (i.e 15 seconds). This duration is known as retention rate. Thus, simple heuristic algorithm is developed to cache videos with high popularity as well as low retention rates.

3) COOPERATIVE CONTENT CACHING

In cooperative caching, multiple MEC servers collaborate with each other to serve the network-wide users' requests. Thus, a single MEC does not need to cache all videos. In fact, different MECs serve the requested videos to any user in the network to minimize the CDN cost and access delay, while maximizing the cache hit ratio. Several works propose cooperative caching in multi-server MEC environment. For instance, in [119] authors proposed to fetch content from neighboring servers using high rate X2 interfaces (instead of typically used S1 interface), when the requested bitrate of the video is not locally available. The scheme aims at improve bandwidth utilization of backhaul links. Despite the fact that cooperative caching can help to efficiently utilize the network resources, storing multiple bit rates of the same video in overlapping regions is still a challenge. Authors in [158] explain the trade-off between caching for high bitrate videos and caching for diversity videos. [158] propose an optimum caching algorithm to store multi-bitrates of videos at the edge servers aiming at maximising the user perceived QoE. The authors adopt the proactive caching policy to update edge caches when the system is idle (i.e., least busy). Video content has huge storage size that makes caching a large number of video files at the edge server challenging; hence, efficient cache replacement strategies are required. To cope with this problem, authors in [159] proposed to improve the performance of mobile video delivery through caching layered videos i.e., instead of caching the entire videos at the edge server, only layers/descriptions of videos can be stored at the edge. Thus when a user request is generated, the edge server can send the cached layer to the users, while in the meanwhile the rest of the video is fetched form the remote/origin server. Similar to cooperation among MEC servers, network users can also implement cooperative strategies to serve contents to each other. A relevant study on users' cooperation in vehicular network is found in [160]. The work proposes a low-cost video caching strategy for mobile devices for delay-tolerant video applications and provided simulation evidences to claim the efficacy. While caching most popular videos at the edge servers significantly improves the performance of video delivery system, there is still a tradeoff between caching high bit rate videos versus caching high diversity videos.

Video retrieval occurs too frequently as compared to cache placement. Hence to enhance the efficiency of edge based caching, the caching policy shall consider the two different time scaled of cache placement and video retrieval. In [161], authors proposed a caching policy that jointly considers the long-term cache placement and short-term cache retrieval in coordinated multi-server system to reduce service delay and cache hit ratio. In [162], authors propose a hybrid caching scheme by combining edge cache sponsoring (ECS) and cellular data sponsoring (CDS). In CDS, the content provider serve free content to the user at the cost of playing advertisements to generate revenue. The user is allowed to select any scheme with the aim to achieve total maximum revenue for both ECS and CDS via cooperation or for an individual provider via competition.

To avoid caching a large number of multiple bitrate versions of the same videos at the edge server, a single high bitrate version of a videos is stored and then transcoded to lower bitrate when lower bitrates are requested by users. However, when the number of user requests increases, the simultaneous transcoding tasks can exhaust edge computation resources. To efficiently uses caching and transcoding resources at the edge server, authors in [163] proposed a RAN-aware adaptive caching scheme, which employs network information acquired from the RAN to estimate the probability distribution of the user requests and uses it to jointly decide video bitrate selection for caching and transcoding. In [164], authors jointly optimize caching, transcoding and retrieval in an energy efficient manner. The probability of uncertain user requests is inferred via historical data to propose a proactive caching policy. The work proposes that edge server can use backhaul links to retrieve requested but not locally available from other neighboring servers.

Few works consider financial metrics (e.g., revenue etc) to design cooperative caching schemes. For instance, in [153], a collaborative online caching algorithm is proposed that minimizes the sum of User Attrition (UA) cost and caching cost for each content. The problem is formulated as Integer Linear Program (ILP). In the online settings, the user requests are revealed one by one to the edge server and the server should take decision before next request to provide the content from local cache (if available), or fetch from neighboring edge servers or from the origin server.

B. EDGE-BASED PROCESSING

Cloud computing has been preferred for computational extensive tasks. However, it has several shortcomings in addition to the common issue i.e., delay constraints as discussed earlier. First, when transcoding a video stream at cloud, multiple versions are created. The number of such transcoded bit rate versions increases the traffic traversing the core network. Second, the cloud-based transcoding strategy is usually fixed and does not adapt to the dynamic changes in viewership. Hence, edge-based processing of video content improves the network resources utilization by sending only the higher bit rate version to the edge server and then the edge server transcodes videos locally as per local viewership. However, edge servers have limited computational resources as discussed earlier and hence must optimize computational tasks to improve the system's capacity. This section presents an overview of research efforts in MEC based video content processing.

1) JOINT CACHING AND PROCESSING

The edge server provides caching and computation resources in close proximity to the end devices. In video streaming applications, mobile devices may need to utilize both resources i.e., fetching cached video content from the edge server and offload computation tasks such as video transcoding to the edge server. For efficient utilization of both resources, the joint optimization of caching and processing functions can improve the network performance to achieve several benefits. In this section, we provide research efforts on joint optimization of caching and processing resource allocation.

The collaboration among multiple MEC servers connected via backhaul links is proposed in [91], i.e., for each new video request, servers collaborate with each other to jointly cache and transcode videos for each other. The idea is that any MEC server in the delivery path (from video origin) to the home MEC server of the requesting user can transcode the video. A major benefit of this scheme is that a server does not need to cache different bitrate versions of the same video, enabling efficient storage across the edge network. Multi-MEC cooperation allows to efficiently utilize overall edge resources in a network. To improve the utilization of a single edge server's resources, end-devices must cooperate to access edge servers. One way to achieve this is efficient scheduling of tasks at the edge server. In [165], authors propose an optimal scheduling strategy for video streaming in best-effort HTTP DASH-based video delivery, leveraging joint coordination among mobile users. Example of joint caching and processing using edge-device cooperation is in [166]. In this work, the authors investigated edge-based processing in VR application. While the VR devices cache most of the components locally, the MEC delivers the components not cached at the VR device. The components are cached and processed at the edge server to reduce computation delay. In order to cope with high communication delay of the component delivery, a task scheduling strategy is proposed. Joint caching and processing has a great significance in adaptive video streaming in which the video bit rate is automatically adjusted (or selected) based on the user device capability (e.g., channel, buffers capacity etc). Instead of storing all possible bit rates of a video, MEC servers decides which bit rates of a popular video are stored and which bit rates are to be generated upon request. For instance, authors in [167] proposed joint caching and processing to implement adaptive bitrate video streaming by formulating it as integer linear program to minimize the the latency of video retrieval. The authors used ML technique to find the video popularity for caching. As caching large number of videos incurs storage cost, caching/processing decision also involves revenue as an important consideration. As a example. authors in [95] studied edge based joint caching and processing to maximise the profit to video service providers. The joint scheme is formulated as a binary optimization problem and solved using multi-armed bandits (MAB) problem and the cache is updated in real time as per users requests. Another interesting approach to solve the joint caching and processing problem is to use a more direct and realistic metric such as the ''number of user requests''. The work in [168] addressed this issue by proposing a joint caching and processing scheme which aims at the maximising the number of service requests that can be served by each base station using a Stackelberg game. The edge server predicts the requests of each BS's users and defines the caching/computing price that maximises its utility. The BS then compete with each other to maximise the availed resources from the MEC at fixed price. A similar approach to [168] is proposed in [169] with additionally considering the user association. The problem is formulated as a mixed integer programming (MIP) to minimize the video retrieval latency in ultra dense heterogeneous networks.

In [170], authors propose a joint video caching and transcoding for VoD streaming called as ''proactive caching and chunk processing (PCCP)''. First video chunks of popular videos based on the user viewing patterns are proactively fetched and stored in neighboring edge servers such that none are replicated. When user requests a video, the respective chunks corresponding to the same video are collected from the neighboring servers and served to the user. A similar approach is used in [119] to fetch only high bitrate versions of a video from the origin/CDN server and transocode to lower bitrate versions at the edge servers. Different edge servers in the neighborhood can collaborate to transcode and share videos with each other via X2 backhaul interface. In [171], authors propose a joint caching and offloading scheme to offload duplicate computation tasks and the requested data content to the edge servers. By jointly optimizing the caching and offloading decision, the scheme minimizes the latency while satisfying the energy consumption of mobile devices. The work uses genetic algorithm to implement an online-learning without requiring future information.

2) JOINT RADIO, CACHING AND PROCESSING

Resource allocation in edge computing can be further improved by considering the user's wireless channel quality in the resource allocation problem. Several works propose the joint optimization of the edge resource allocation and the user's communication resources to meet the QoS requirement. Some representative works are listed [137], [172], [173], [174], [175], [176].

In [172], authors proposed a joint radio, caching and processing scheme for MEC-assisted VR applications. The edge server proactively caches some parts of the videos and process these videos. The caching and processing resource allocation at the edge is jointly optimized with the transmission rate constraint of end VR devices to maximize average delay, while guaranteeing transmission rates, cache size, energy consumption and front-haul capacity. In [173], MEC-assisted VR delivery is proposed in which some parts of the videos are cached at the edge whereas others are stored at the VR device. The algorithm jointly optimize caching and computing resources to determine the parts of videos (more specifically called Field of Views or FOVs) to be cached and processed at the mobile device and which ones to offload to the edge server under three constraints (i.e., cache size, power consumption and latency). Due to the more dynamic and complex nature of the problem, advanced techniques based on DL are being proposed. For instance, in [174], the joint caching, transcoding and transmission of videos is formulated as Markov Decision Process (MDP) and solved using Deep Reinforcement Learning (DRL) to improve user experience in adaptive video streaming. In [175], authors consider joint optimization of caching, processing and radio resources to maximize system revenue. In [176], authors propose video caching and transcoding scheme in heterogeneous virtual MEC networks. The scheme jointly optimize caching and transcoding with the network radio conditions during the cache placement and delivery phase respectively. In [137], authors propose to perform the pre-processing of the videos at edge servers using a lightweight Deep Neural Network (DNN) model, and upload the results to cloud nodes for further analysis to produce a complete video analytic solution. The aforementioned state-of-the-art discussed in this section is summarized in Table 6.

C. LESSONS LEARNED AND CHALLENGES

The main objective of deploying video streaming applications in the mobile edge networks is to minimize the perceived delay, the network cost and the energy consumption, and to maximize different resources utilities (e.g., computation, memory, and bandwidth) and the users' QoE. Hence, most of the recent works in the literature focus on establishing a joint optimization to balance different purposes. The MEC strategies are classified into centralized and collaborative video caching and processing. The centralized approach relies typically on a base station or eNodeB to deploy different tasks. In general, this strategy is simpler and the optimal resource decisions are less complex. On the other hand, the joint strategies involve multiple MEC entities that collaborate to offload the streams, while respecting the objective constraints.

More specifically, when the users' demand increases, MEC servers can collaborate to serve content to its neighboring servers. The collaboration among MEC servers becomes essential when storing all requested video content locally is not possible due to limited storage capacity. Typically, the caching and processing of content is considered as a joint problem in various works to improve the overall network performance. Moreover, due to its strong impact on the offloading capability, the channel quality is also considered in the joint optimization problem. In this context, previous works proved that the optimal solution is NP-hard and approximation algorithms have to be designed.

Due to the large size of videos and increasing use of video-based services, traditional cloud-based caching schemes such as MPV, LRU and LFU schemes are inefficient to meet the network demand. MEC-based video caching can benefit from machine learning to implement proactive content caching at the network edge to reduce latency and improving the utilization of limited caching capacity.

VI. DEVICE-TO-DEVICE (D2D) COOPERATION IN MEC SYSTEMS

Over the last decade, a significant increase in the computational power in the end devices has been observed. These resource-rich devices can collaborate with each other via D2D communication to achieve network-wide performance gains. In the MEC system, mobile devices can implement

TABLE 6. State-of-the-art of MEC in video streaming.

Category	Ref	Description of Contribution	Evaluation Metrics						
			Delay	Energy	Hit ratio	Network cost	Capacity	Revenue	Other
Caching and Proactive Content Fetching	[146]	Fetch Most Popular Videos (MPV) using nation-wide popularity.	✗	✗	✗	✗	✓	✗	✗
	[153]	Proactively fetch contents to reduce the sum of storage cost and user attrition cost.	✗	✗	✗	✓	✗	✗	✗
	[157]	Caching based on popularity and retention rate of video streams to maximize their video bitrate.	✗	✗	✗	✗	✓	✗	✗
	[152]	Smart caching based on user behavior to maximize the hit rate of contents within limited time.	✗	✗	✓	✗	✗	✗	✗
	[159]	Cache layered videos for multiple user groups. Users in same group share caching cost.	✓	✗	✓	✗	✗	✗	✗
	[131]	Prefetch popular videos from YouTube using ML-based prediction.	✗	✗	✓	✗	✓	✗	✓
	[163]	Cache videos that serves the maximum users by estimating the requested qualities using RAN information.	✓	✗	✓	✗	✓	✗	✗
	[177]	Edge caching strategy for vehicular networks in which video content is stored on the RSU and fetched by users.	✓	✗	✓	✗	✗	✗	✗
Cooperative Caching	[119]	Prefetch highest bitrate version of the video from CDN server using X2 interface.	✓	✗	✓	✓	✗	✗	✗
	[160]	Prefetch video chunks into playout buffer from encountered vehicle caches/ stream from the cellular.	✗	✗	✗	✗	✗	✗	✓
	[158]	Solve the multiple bitrate video caching problem using polynomial complexity algorithm.	✗	✗	✗	✗	✗	✗	✓
	[161]	Operators advertise contents to attract a mobile user to request cached contents instead of non-cached contents.	✗	✗	✗	✗	✗	✓	✗
	[162]	Context-aware adaptive caching through network virtualization at the edge.	✗	✓	✗	✗	✗	✗	✗
	[164]	Jointly optimize caching of bitrate-aware files and the scheduling requests to minimize energy consumption.	✗	✓	✓	✗	✗	✗	✗
	[91]	Jointly optimize caching and transcoding resources using ILP to minimize the backhaul network cost.	✓	✗	✓	✓	✗	✗	✗
	[178]	Cooperative caching when users' preference is unknown and only the historical content demands.	✓	✗	✓	✗	✗	✗	✗
	[179]	Joint caching placement (long-term) and video retrieval (short-term) in coordinated multi-server system.	✓	✗	✓	✗	✗	✗	✗
	[180]	Using Q-learning, find the appropriate cache state to improve caching mechanism.	✗	✗	✗	✗	✗	✗	✓
	[167]	Joint collaborative caching and processing at multiple servers to minimize the delay of video retrieval.	✓	✗	✓	✓	✗	✗	✗
	[170]	Cache only the video chunks to be watched and collaboration among neighboring MECs to improve utilization.	✓	✗	✓	✓	✗	✗	✗
	[181]	A greedy collaborative caching strategy in heterogenous MEC network to minimize total delay of all users..	✓	✗	✓	✓	✗	✗	✗
Cooperative Processing	[137]	Edge nodes pre-process video data using DNN model and upload the results to cloud for further analysis.	✓	✓	✗	✗	✗	✗	✗
	[182]	Peer-offloading using perceptual parameters such as pausing frequency, watching percentage and bit rates.	✗	✗	✗	✗	✗	✗	✓
Joint Caching and Processing	[166]	Reduce communication-resource consumption by using MEC to compute components not stored on VR device.	✗	✗	✗	✓	✗	✗	✗
	[165]	Implement per-channel optimal service-aware resource allocation.	✗	✗	✗	✗	✗	✗	✓
	[167]	A method to decide (i) transcode locally or (ii) fetch from neighboring MEC or (iii) fetch from the origin server.	✓	✗	✓	✓	✗	✗	✗
	[95]	A video caching and processing model that offers maximized profit to video service provider.	✗	✗	✗	✗	✗	✓	✗
	[183]	A joint caching and transcoding scheduling strategy to minimize the energy consumption.	✗	✓	✗	✗	✗	✗	✗
	[168]	BSs compete to maximise its revenue by maximising the request rate that can be served by the MEC server.	✗	✗	✗	✗	✗	✗	✓
	[169]	Minimizes the average retrieval latency of all users using proactively caching and user-BS association scheme.	✓	✗	✓	✓	✗	✗	✗
	[170]	Proactively cache video chunks likely to be watched instead of whole video content.	✓	✗	✓	✓	✗	✗	✗
	[119]	Online bit rate conversion to the requested version of videos fetched from the origin/CDN servers.	✓	✗	✓	✓	✗	✗	✗
	[171]	Joint task offloading and dynamic caching strategy to reduce overall latency of all mobile devices.	✓	✓	✗	✗	✗	✗	✗
	[184]	Joint optimization of caching and transcoding for resource allocation in DASH.	✗	✗	✓	✗	✓	✗	✓
Joint Radio, Caching and Processing	[172]	To maximize the average tolerant delay while meeting the rate constraint in VR devices.	✓	✗	✗	✗	✗	✗	✗
	[173]	To decide whether to pre-cache (and if yes, which) parts of the field of views (FOVs), with local pre-processing.	✗	✗	✗	✓	✗	✗	✗
	[174]	Jointly considers buffers, video quality, edge caching, transcoding and transmission for energy saving and QoE.	✗	✓	✗	✗	✗	✗	✗
	[175]	Jointly optimize the caching, radio and resource allocation to maximize the system revenue.	✗	✗	✗	✗	✗	✓	✗
	[176]	Joint multi-bitrate caching and transcoding by allocating physical and radio resources based on network stats.	✗	✗	✗	✗	✗	✓	✗
	[185]	Jointly considers transmission, coding, caching, and computation to meet delay requirement in mobile VR.	✓	✗	✗	✗	✗	✗	✗
	[186]	Jointly optimize cachig, processing and transmission to reduce latency.	✓	✗	✗	✗	✗	✗	✗

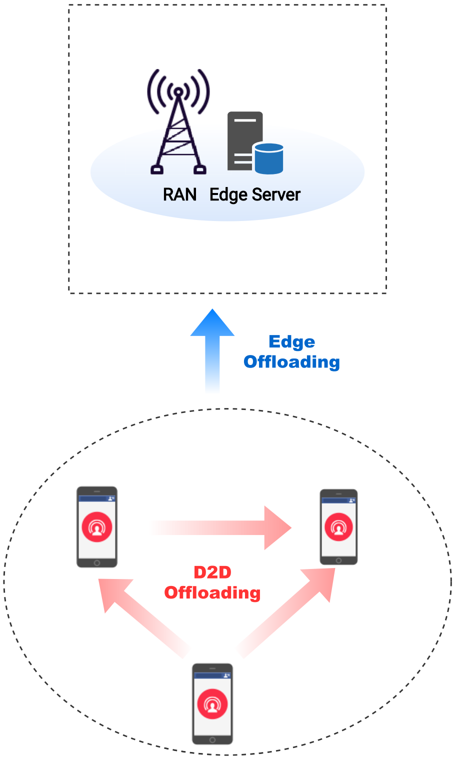

FIGURE 10. **D2D cooperation in MEC systems.**

D2D communication to achieve several benefits: First, to alleviate the computation load on the edge server, particularly if the edge server has limited resources or high computation load by offloading tasks to mobile devices. Second, devices can collaborate to exploit and reduce communication latency by exploiting the high speed D2D communication. Third, due to the short range communication, the device's energy consumption can be significantly reduced. Fourth, the spectrum utilization of cellular network can be multiplexed to improve system's overall capacity.

Both D2D and MEC systems aim to benefit from the proximity of the devices to achieve performance gains. In D2D networks, a device connects with another nearby device to communicate at smaller distance and thus achieve high data rates and less latency. In MEC systems, devices can benefit from the closely located edge servers to fetch content and offload computations thus achieving low latency and computational gains. Thus, the integration of both systems will intuitively allow network users to benefit from both technologies in a range of applications. Figure 10 illustrates D2D offloading in MEC systems.

A. RELEVANT WORKS

D2D communication has been used in wireless networks to gain several benefits such as energy saving, less bandwidth utilization and enhanced quality of experience [187], [188], [189]. For instance, [187] proposes a generic framework for D2D communication to improve video quality and reduce energy consumption and bandwidth utilization. In [189], authors propose Wi-Fi Direct based D2D scheme to scale the network size and increase throughput.

Recently, D2D-enabled MEC systems have been proposed [112], [114], [190], [191], [192], [193], [194], [195], [196], [197]. In D2D-enabled MEC systems, mobile devices can not only exploit the communication resources over D2D links but they can also benefit from the computational resources of under-utilized devices. Every D2D enabled device in the network is a computational resource and thus can aid to improve the overall computational capability of the system by realising cooperation among devices and edge server. Such kind of D2D-MEC cooperation can be extremely useful.

Several works have recently proposed the integration of D2D communication in MEC systems to achieve performance gains. In [190], authors proposed that D2D communication can be used by mobile users to find alternative resources when accessing the cloud server encounters long delays due to intermittent wireless connectivity. D2D-fogging is proposed in [112], a concept similar to standard edge computing architecture with D2D communication enabled among user devices. The D2D devices collaborate with each other to share their computation resources controlled by the base station (associated edge server). The tasks are offloaded online, aiming at minimizing the time-average energy consumption. In [193] authors proposed a D2D-ECN (Edge Computing Network) framework for computation offloading. Using D2D-ECN, computation intensive devices can offload tasks to resource rich devices. Q-learning has been used to perform optimal resource allocation in a point-to-point offloading system. In [113], authors proposed D2D-MEC system in which devices offload their computation to nearby idle devices (helpers). The task offloading is jointly optimized with communication and energy beamforming to maximize the sum-computation rates of users. In [192] authors used D2D communication in MEC system to maximize the number of devices supported by the system with communication and computation constraints. A mixed integer programming (MIP) formulation is presented for the D2D-MEC system and solved by decomposing the MIP problem into two sub problem. The simulation results reveal significance of the proposed D2D-MEC system in cellular networks. In [114], authors highlighted the use of D2D collaboration in MEC system to improve the overall system capacity. Computing in D2D network is more complicated when it comes to resource and topology management. The work in [198] presents resource management in D2D-MEC system such as link selection and subchannel allocation, transmit beamforming, transmit power and receiver combiner.

In [56], authors investigated experimentally the feasibility of video transcoding at the user devices. The authors transcoded videos of short duration on mobile phones to measure the transcoding time. In [192], authors propose a D2D-MEC system to improve the computation capacity of the network i.e., to maximize the number of devices supported by the cellular networks. In this scheme, a device can offload its tasks to an edge server and a nearby D2D device. In [11] a joint communication and computation optimization scheme for MEC is proposed that leverages on D2D communication. The nodes use collaborative beamforming to wirelessly charge other devices and offloading computational tasks to idle devices. In [115], authors studied D2D-enabled multi-MEC system in which a user offload computational tasks to multiple local users using time division multiple access (TDMA) transmission to achieve latency reduction. The timeslots are divided into three phases. In the first phase, the task is offloaded to the helper nodes. In the second phase, the helper node executes the task. The computation results are downloaded in the third phase.

Quantile-based CSMA [199] and ITLinQ [200] have been proposed to improve the network throughput for video transmission in static networks without consideration for dynamic user mobility. In [209] and [210], the authors proposed the caching at the network edge to reduce the congestion and the delay transmission for video content. The caching technique is also proposed in [211] and [212] where the D2D communication is used to create a direct communication between nearby mobile users. These caching solutions are effective for stored video but they introduce higher throughput, delay, and jitter in case of live streaming. In [205], the authors proposed FlashLinQ based on CSMA protocol to establish a D2D link. The links are classified according to the link's priorities where the links with higher-priority do not suffer from significant interference with lower priority links. The algorithm uses the priority for each link to schedule its activation time. FlashLinQ has a good performance in terms of latency; however, the video quality-aware mechanism was not considered in this work and therefore its suitability for D2D on-demand video streaming remains open. D2D communication has been proposed for video streaming in [206]. The proposed model improves the resource utilization in 5G by using a scheduling algorithm for effectively sharing the multimedia content using D2D communication. However, this algorithm studied only the case for one user and one video without considering the case for a multi-user with multi video sources. In [207], D2D multicast communications for live streaming video is investigated. The proposed scheme uses the frame priority (FP) to improve the QoS perceived by users and the users' satisfaction with the video quality. The proposed model considers the encoding characteristics of video streaming and users' feedback to ensure that the frames to be retransmitted are valuable for decoding.

Table 7 summarizes the recent research studies on D2D-enabled MEC systems.

B. LESSONS LEARNED AND CHALLENGES

Device-to-device collaboration for communication is an efficient technique and is widely proposed in wireless networks. D2D standard protocols also exists for direct communication among mobile users. D2D communication is used for improved network coverage, higher throughput and reduced energy consumption. In MEC, D2D has been proposed to improve the offloading capacity of the edge server. Particularly, devices can offload computational tasks to helper devices instead of the edge server. Such D2D offloading can be efficient when the total delay (sum of computation of communication delays) for local processing or edge-based processing is higher then D2D processing. D2D-offloading can also be used on the downlink i.e., when the edge server is overloaded, the server can offload user's tasks to the resource-rich helper devices.

Incentive mechanisms need also to be envisaged to systematize the revenue between broadcasters, MEC infrastructure and D2D-participants. Furthermore, the security and privacy aspects of the D2D offloading for video crowdsourcing applications are not well considered in the literature and they are still the major limitation for such an approach. In this context, blockchain-based techniques have been proposed as a strong tool to handle an incentive and trusted environment for many applications. However, it is rarely studied for edge-assisted video streaming systems. We believe that blockchain combined with distributed and cooperative network can guarantee data integrity and effectively realize a trusted topology, particularly in 5G networks. Moreover, since blockchain provides limited data storage, efforts should focus on establishing a balance between off-chain and on-chain strategies.

VII. INTELLIGENT EDGE FOR VIDEO STREAMING

Both Machine Learning (ML) and edge computing are making profound impact in several domains, however when they are combined, they can bring a more intriguing user experience. The intersection of both is being considered as a suitable platform for vertical applications e.g., automotive (e.g., autonomous cars), healthcare [209] (e.g., remote surgeries, real-time assistance),security [210] (e.g., intrusion detection), manufacturing (e.g., predictive maintenance), retail (e.g., VR-enabled online shopping experience and personalized suggestions) and connected homes (e.g., temperature control, smart smart doorbells, access control, and smart lighting). Most of the video streaming services that benefit from edge computing are using machine learning or have the potential to benefit from it.

Deep learning (DL) is a sub-domain of machine learning which has been successful in solving highly complex problems. However, to meet the computational requirements, DL applications typically leverage cloud computing. An associated drawback of running deep learning on the cloud is the that data might need to be transferred to the remote cloud server [211]. If such data is not available at the cloud (e.g., data coming from visual sensors such as IoT video sensors) [212], this "can result in network congestion and delays due to the long-distance or multi-hop transmissions" [213]. Two illustrative examples of scenarios which involves transmission of videos to a server for processing include object/event detection (e.g., automatic fire detection, crime detection) using videos from surveillance cameras, and

TABLE 7. Device-to-Device (D2D) cooperation in MEC systems.

Category	Ref	Algorithm	Evaluation Method	Metric	Description
Methods and Use cases	[190]	Numerical model	Not Available	Task success rate	Propose (i) optimal and (ii) periodic scheme to offload tasks and retrieve results.
	[112]	Lyapunov optimization	Simulation	Energy	Dynamic D2D offloading that minimize energy consumption of all users, while preventing over-exploiting helper device's resources
	[192]	MIP	Simulation	Capacity	Maximize the number of devices supported by cellular network by minimizing edge resource computation and optimal D2D pairing.
	[114]	Lagrangian Multipliers	Simulation	Capacity	Propose a file allocation algorithm in D2D MEC systems to improve the caching capacity
	[208]	Convex optimization	Simulation	Energy	Collaborative beamforming to wirelessly charge other devices using D2D and offloading computational tasks to idle devices.
	[115]	MINLP	Simulation	Latency, capacity	User offload computational tasks to multiple local users using time division multiple access (TDMA).
Resource Optimizing	[113]	Convex optimization	Simulation	Capacity	Jointly optimize energy beam-forming, communication and computation resources to maximize sum of computation rates of users.
	[193]	Q Learning	Simulation	Latency	Jointly optimize computational offloading, power allocation and CPU frequency adjustment for computation offloading and resource management.
	[198]	NLIP	Simulation	Energy	Select links in D2D to minimize energy consumption and then jointly optimize resource allocation and MIMO signal design.
Performance Evaluation	[56]	Android testbed	Real data	Capacity, Energy	Transcoded videos of short duration on mobile phones to measure the transcoding time.

driving assistance in autonomous driving (e.g., sending captured videos from preceding vehicles to the server). To cope with this challenge, edge computing comes as a suitable choice. By processing such data at edge servers, the latency is reduced and most of the data required to run the DL models is either available at the edge or at closely located end devices, hence improving bandwidth utilization.

One of the broad applications of DL for multimedia streaming is the video analytics for surveillance systems, which was introduced in the previous section IV-E. In fact, we showed that some monitored areas are very critical and need continuous surveillance, such as military borders. In this scenario, video frames (or video streaming) are sent instantaneously to remote servers for real-time object/accident detection at every small interval, which incurs huge data transfer. This type of applications has also stringent latency constraint, as it requires a prompt intervention, if an accident is detected.

Furthermore, for a better accuracy of the results, high-resolution videos having high data volumes are transmitted by the source to the computation units. The accuracy is also affected by the speed of feeding the data to the model, which requires bandwidth availability and efficient communication technologies to follow the requirements of the DL model. It is worth to mention that video surveillance cameras are highly pervasive. For example, 170 million cameras are in deployed in China roads only [214]. These devices are generating a huge amount of data as previously described. It is reported that the data produced by IoT devices will witness a growth of 28% by 2025 [215], where 65% of it is related to the surveillance cameras [216]. Even-though the conventional wisdom resorts to the centralized cloud servers for analytics owing to their high computational capacity, offloading these bulk of data presents scalability issues, as the access to the cloud can experience bandwidth bottleneck when the number of video sources increases. Moreover, due to the strict latency requirements, the cloud is no longer a feasible solution.

Recently, DRL techniques have been applied for video streaming and network selection in [225], [226], and [227]. In [217], a system that generates adaptive bitrate (ABR) algorithms using reinforcement learning (RL) is proposed. This model trains a neural network model to select bitrates for future video chunks based on observations collected by client video players. In [218], the authors proposed QARC (Quality Aware Rate Control) algorithm to obtain a higher perceptual video quality with possible lower sending rate and transmission latency. QARC uses DRL algorithm to train a neural network for selecting future bitrates based on previously observed network status and past video frames. In [219], a predictive panoramic video delivering based on DRL has been proposed. The LSTM model is used to predict the user's field of view (FoV) in the next few seconds. These systems focus on the continuous prediction of video resolution and average video bitrate.

In the following, we provide a more detailed review of the research contributions on the use of ML/DL techniques in MEC systems for video streaming use cases. The section covers various areas of video streaming i.e., proactive caching, optimal computation offloading, adaptive streaming, video analytics and AR/VR streaming. A summary of these works is presented in Table 8.

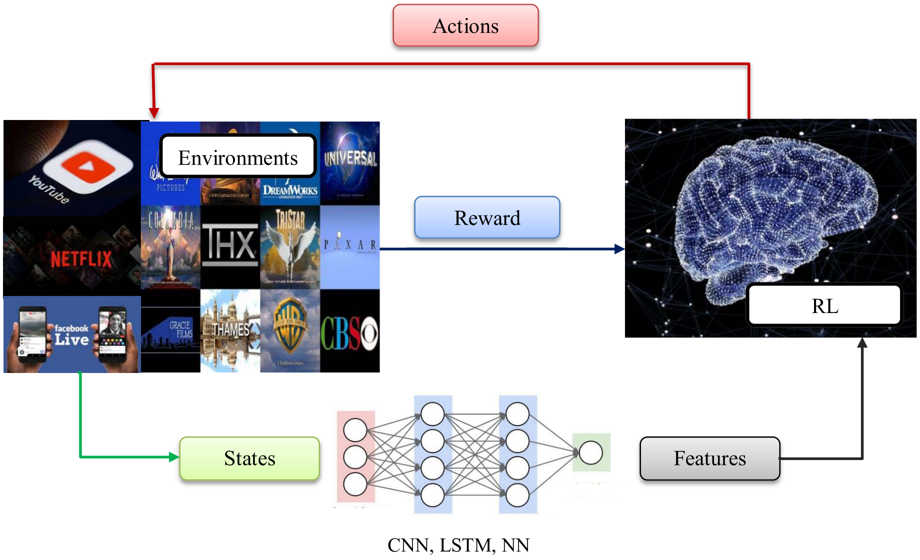

FIGURE 11. DRL for video transmission.

A. PROACTIVE CACHING

Machine learning is being widely proposed to implement proactive caching. Proactive caching is extremely useful when the user viewing patterns are not constant and resource allocation is required to meet service requirements. Machine learning thus brings new possibilities by implementing intelligent edge caching to capture hidden features from historical data or current network stats. For instance, ML techniques can be used to forecast video popularity or predict future user requests based on historical viewing patterns of users. ML assisted proactive caching schemes are investigated in [149], [228], [229], [230], [231], and [232]. In [149], authors integrated three ML techniques i.e., Auto Regressive Integrated Moving Average (ARIMA), Multiple Linear Regression (MLR), and k-Nearest Neighbor regression (kNN) to predict the users social patterns and used it for caching resource allocation. Advanced prediction techniques such as Long Short-Term Memory (LSTM) deep learning model can be used for more efficient prediction of video popularity. LSTM has been effectively used in [223] to predict both long-term and short-term video content popularity. While historical user viewing data can provide a reasonable estimate of future behavior, other parameters can also be considered for effective caching policies. In [221], authors applied Deep Reinforcement Learning (DRL) to implement caching scheme by using user requests, network constraints, and external information. In the proposed DRL-based scheme, the inputs are the index of currently requested content, the number of requests for the cached content, and the time when the cached content was most recently requested. Using these inputs as ‘‘states’’, the DRL agent determines whether or not to cache the currently requested content. The reward function is the offloaded traffic in each request. The DRL model evaluated using the MovieLens dataset [236] shows improved performance against traditional caching policies i.e., FIFO, LFU, and LRU. In [148], a network-aware caching scheme is implemented using machine learning. The users' behavior and networks parameters such as cache size, bandwidth, and load are to build a neural network model to estimate video popularity. The estimated video popularity of video content is used in mixed integer programming to calculate cache size and content placement. Authors in [222] developed a DQN model to implement a mobility-aware joint caching and computing design in vehicular networks. The model states include parameters i.e., available caches, available vehicles, contacts per time slot, and contact times for every vehicle. The agent performs the action i.e., decides which server and/or vehicle is assigned to the requesting vehicle, and determines whether to cache the requested content or not. The reward function is cost of communication for data transmission.

B. COMPUTATION OFFLOADING

The computation offloading and resource allocation at edge servers is often solved as an optimization problem using integer programming or heuristics [154], [170], [237]. However, recently deep learning models such as DRL and Deep Q-learning networks (DQN) are proposed to solve resource allocation in various networks. For instance, in [224], DRL has been employed to jointly optimize caching, computation offloading policy, and radio resource allocation to minimize the average end-to-end delay. The DRL model in this work uses state parameters including number of requests, number of fog nodes, task sizes, computation requirement, popularity of requested content, and channel quality vector between devices and fog nodes. The action space includes multiple decisions such as the fog node to serve the user, whether to store the requested content or not, the allocated communications sub-channels and computational resources. The reward

TABLE 8. Machine learning in mobile edge computing for video streaming.

Category	Ref	Model	Method	Dataset	Description
Caching	[149]	ARIMA, MLR, KNN	Testbed	Real	Forecast video popularity pattern to cache popular videos
	[220]	Q-learning	Simulation	Synthetic	Distributed cache replacement using Q-learning based on content popularity
	[148]	ELM	Simulation	Real	Estimate the unknown popularity of contents for caching.
	[221]	DRL	Simulation	Real	Learn end-to-end caching policy using DRL
	[222]	DQN	Simulation	Synthetic	Estimating mobility-aware reward for joint caching and computing in edge.
	[223]	LSTM	Simulation	Real	Cache videos using long-term and short-term popularity prediction of videos.
Computation Offloading	[224]	DRL	Simulation	Synthetic	Minimizes end-to-end delay for caching, offloading, and radio resources
	[225]	DQN	Simulation	Synthetic	Minimizes energy, computation, and delay cost for multiple task offloading
Adaptive Video Streaming & QoS	[226]	RBM	Simulation	Real	Using RBM with linear classifier for concurrent transmission to multiple users with QoS guarantees
	[227]	DRL	Simulation	Real	Use DRL to assign users to the appropriate server in multi server MEC system.
	[228]	DRL	Simulation	Synthetic	MEC-assisted adaptive video streaming for driving assistance for improved "quality level switching".
Video Analytics	[229]	CNN	Testbed	Real	Edge-based faster object detection in videos.
	[230]	CNN	Testbed	Real	Edge-based object recognition model for food items.
	[231]	CNN	Testbed	Real	Improving frame rate and accuracy with edge computing running CNN.
	[232]	CNN	Simulation	Real	Collaboration among user and edge server to train a CNN Model to improve object recognition.
	[233]	L-CNN	Simulation	Real	A CNN model for resource-constrained edge server.
	[234]	ADMM	Simulation	Real	Latency-aware video summarization using ADMM, to understand the story-line of a video before a client requests the complete video content summarizing
	[235]	DNN	Testbed	Real	Automated ML framework using DNN to perform object recognition in device-captured frames sent at the edge server, while improving recognition accuracy.

function is the weighted sum of time for computation and content delivery. In [225], the edge resource allocation problem considering multiple tasks offloaded to the edge server by mobile devices is investigated. The problem is formulated as mixed integer program and then solved using DQN. The system state includes all users' offloading decisions and bandwidth allocations for all users. The action is an index that denotes the selection among two different neighboring states. The reward is chosen as 1 if the total offloading cost is decrease after taking an action and −1 if the cost is increased. If the offloading cost remains constant after an action, reward is 0. The total offloading cost is the sum of energy cost, computation cost, and delay cost.

C. ADAPTIVE STREAMING AND QoS

Adaptive streaming is the ability of the video delivery system in which the video quality is automatically adjusted according to the user's preference or network conditions. More specifically, the video bitrate is automatically adjusted as the user's channel quality degrades or improves, or the user requests a particular bitrate. Although mobile edge computing provides extra support to implement adaptive video streaming in video services, recent development in deep learning particularly deep reinforcement learning are being proposed to improve MEC based adaptive streaming. In [226], authors proposed a multi-stage learning system using Restricted Boltzmann Machine (RBM) to manage simultaneous video transmission which guarantees a minimum quality level for each user. In [227], authors used DRL model to assign users to the most suitable edge server aiming at improving quality of service in video streaming. In this model, the states include resource usage, viewer scheduling information, and current viewer request. The agent's action is to assign a viewer's request to a server. When a new user's request is assigned to a server, it will increase the load on the system which impacts the QoE for the new users. Thus, the reward function is chosen as the opposite number of the overall penalty for the coming viewer. The aforementioned works on adaptive video streaming do not include mobility scenario such as vehicular users. To implemented MEC-assisted video streaming services in road scenarios, authors in [228] propose the use of DRL technique. The states of the DRL model include video quality level, queue length in the edge server, and average wireless channel gain. The wireless channel gain captures the mobility pattern of users. The agent uses the states to decide the action i.e., assigning the video quality levels to users. For each action, the reward is computed i.e., weighted sum of high quality level assigned to user, negative reward of quality level switching and the negative reward for queue length.

D. VIDEO ANALYTICS

Video analytics are advanced services using streaming videos to extract real-time insights. There is a broad range of applications which requires real-time analytics in streaming services such as surveillance [229], autonomous vehicles, cognitive assistance [230], video summarization [234], and filmography etc. However, such analytics are only useful when provided in real-time i.e., within strict delay deadlines. These advanced services rely on deep learning techniques which requires huge computational resources. MEC is being increasingly proposed as a promising solution to perform faster analytics and at lower data transmission delay. Authors in [229] used CNN model for object detection in edge computing based video analytics for real-time surveillance and achieved accurate Region Of Interest (ROI) detection. The edge servers train local models on data collected from devices connected to it in a distributed way whereas the cloud server contains a global model which is shared with the edge servers. In [230], CNN model is used for food object recognition for latency-sensitive application using edge computing services. The recognition latency and energy consumption of devices is reduced by distributing the model learning tasks among edge servers and the cloud server. The initial image pre-processing such as identification of blurred images is performed at devices using simple methods i.e., K-means algorithm. Then image segmentation i.e., extracting background and foreground object from the image is performed at the edge layer. Lastly, the DNN model i.e., CNN for object recognition is executed at the cloud. In [231], a CNN model, Yolo [238] is implemented on edge server to support computationally weak front-end devices by offloading their tasks to the edge to achieve a higher frame rate and accuracy. The scheme estimates the network conditions and the actual requirements of the application to decide the offloading strategy in CNN computing. DeepCham, a DL-based object recognition model is proposed in [232] using collaboration among users and edge server. Devices and edge servers collaborate to train the CNN model to improved accuracy of the model. Similarly, a lightweight CNN model is developed in [233] for resource-constrained edge server with reduced numbers of filter numbers in each layer.

Video summarizing on edge server and sending the video summary instead of the whole video to let the user understand the video content have many attractive use cases. User experience can be significantly improved if the video summary size is tailored according to the varying network bandwidth and user viewing behavior. Authors in [234] proposed latency-aware edge computing to achieve the adaptive summarization with improved bandwidth utilization and viewing experience. In [235], authors applied DNN to automatically perform object recognition in videos. The proposed method improves the accuracy of the object recognition while meeting the frame capturing rate constraint.

E. AR/VR

MEC is being used in AR/VR streaming services to accomplish DL tasks such as object recognition and features extraction at much higher rates. This involves relatively more complex tasks such as field of view (FOV) prediction in in 360° using deep learning [239], [240], [241]. The predicted FOV is used to determine the spatial region in video to be fetched from the VR content provider. Edge computing can help to improve such applications by reducing the latency as illustrated in [242]. In Table 9, we have provided a list of useful datasets of video streaming with brief descriptions and applications for prospective researchers.

F. LESSONS LEARNED AND CHALLENGES

Machine learning techniques can be efficiently applied in mobile edge computing to gain several benefits. In video caching, ML techniques can predict the cache demand to allocate storage resources and prefetch video content before users request them. User requests can be accurately predicted using user's viewing patterns or video popularity on both short-term and long-term basis. LSTM and DRL techniques are relatively efficient for proactive video caching. For computation offloading and resource allocation on the edge server, DQN and DRL have been widely proposed. In addition to these, MEC also provides a platform to deploy ML-based services for several applications including video analytics, interactive videos, and content detection. A detailed analysis of the ML-based techniques discussed in this section shows that MEC improves the application performance in several ways; (i) perform training and inference of DL models fully or partially, (ii) MEC can be used for pre-processing of huge amount of data, (iii) collaboration among edge servers and cloud improve system's capacity. Such applications (e.g., VR and AR.) do not only require a successful classification of the objects, but also a high accuracy rate. However, the stringent requirements in terms of latency halt them from computing remote tasks. In other words, even 100 ms end-to-end latency of edge classification may no longer match the highly variable location of moving objects [252]. In this context, authors in [253] expected a latency equal to 20 ms to avoid motion sickness in virtual reality applications. Compression techniques to minimize the data transmission latency are proved to be promising [252]. However, compression

TABLE 9. Datasets for video streaming research.

Dataset	Ref	Year	Description	Applications
Facebook Live	[243]	2018	The dataset consists of a list of 1.506.473 videos collected from Facebook live over a period of 34 days. The videos are published by 408.231 different broadcasters.	Multi-cloud resource allocation, edge caching, and traffic allocation problem.
YouTube-8M	[244]	2019	The original dataset contains 6.1 Million video IDs, 350,000 hours of videos and 3862 classes. The average number of machine-generated labels for each video are 3. The extended version of the dataset contains human-verified labels on about 237K segments (average of 5 segments/video) on 1000 classes from the validation set.	Video analytics such as activity recognition, captioning, and classification.
Twitch dataset	[47]	2015	A dataset of crawled data collected from Twitch every five minutes in a one-month period. The dataset consists of 11 parameters e.g. number of viewers, number of active viewers, followers gained and parameters of particular streamer.	User-generated live streaming
VR Streaming dataset	[245]	2017	The dataset contains head tracking of 48 users (24 males and 24 females) watching 18 spherical videos from 5 categories.	User behavior patterns in VR streaming, visual attention modeling, gazing prediction, user identification based on head motion pattern.
360° Videos dataset	[246]	2019	A dataset of 28 360° videos based on the taxonomy, and recorded viewport traces from 60 participants watching the videos.	Viewing experience in 360° video streaming
LIVE Mobile Stall Video Database-II	[247]	2017	The dataset contains a total of 174 distorted videos generated from 24 reference videos with 26 unique stalling events. It also includes both continuous-time and overall subjective scores from 54 unique subjects.	QoE predictive models, performance benchmarking, and user-centric mobile network planning
YouTube's mobile video streaming dataset	[248]	2019	Dataset is acquired for 783.88 hours of playback time and contain several network and traffic related parameters i.e. type of measurement, timestamp, scenario, videoID, packet arrival timestamp, packet length in bytes, source and destination IP address, source and destination TCP port and the payload protocol number, TCP flow duration, flow size and initial video playback delay.	Studies on mobile videos initial delay.
5G trace dataset	[249]	2020	The dataset is generated from two static and moving cars, using two different applications i.e. video streaming and file download. The dataset contains client-side cellular key performance indicators (KPIs) e.g. channel-related metrics, context-related metrics, cell-related metrics and throughput information.	Video streaming QoE in 5G networks
UVG dataset	[250]	2020	Dataset for Ultra Video Group (UVG) video codec analysis, consisting of 16 versatile 4K video sequences, captured either at 50 or 120 frames per second (fps).	Subjective and objective quality assessments of next-generation VVC codecs.
LIVE-NFLX-II	[251]	2018	A subjective Video QoE dataset which includes 420 videos that were evaluated by 65 subjects. These videos were generated from 15 video contents streamed under 7 different network conditions and 4 client adaptation strategies.	QoE studies e.g. video quality fluctuations, rebuffering events of varying durations and numbers, spatial resolution changes, and diverse bitrate/quality levels and video content types.

strategies should be revisited to establish a trade-off between data transmission and classification accuracy. Moreover, edge-assisted on-device inferences are envisaged to avoid remote transmissions.

VIII. FUTURE RESEARCH DIRECTIONS

A. MOBILE EDGE COMPUTING IN 5G/6G NETWORKS

The current 5G networks have the potential to provide a robust network for connecting people around the world. However, challenges persist as new services with more stringent requirements are introduced. The next generation of wireless communication known as 6G [254], [255] can meet these unprecedented requirements that 5G might not support. In fact, when we deep dive on the roadmap of the 8K, VR, AR, and even eXtended Reality (XR), we can point that it not possible with 5G to longer support these volumetric videos. This is due to the 5G design and baseline KPIs that aim to reach an average ubiquitous capability equal to 100 Mbps downlink and 50 Mbps uplink [256]. Whereas, the long term roadmap of eXtended reality videos is to be pervasive everywhere on a network offering multi-gigabytes to users. Moreover, current 360° 4K videos need 10 to 50 Mbps to be delivered to devices, meanwhile the next generation 8k videos require from 50 to 200 Mbps, which cannot be served by 5G. Also, looking to the futuristic types of videos, such as 6 Degrees of Freedom streamings, and holography presenting full immersive experience and requiring few Gbps to Tbps, 5G network is highly challenging. Aside from the incapacity to deliver a fast data rate for volumetric applications, 5G is not designed for ultra-precision streaming tasks, such as remote surgeries. On this basis, future works on 6G paradigm should think about enhanced Ultra-Reliable and Low-Latency Communications (URLLC) services, advanced massive machine type communications (mMTC) services, and eventually new network capabilities that allow to deploy pervasive streaming on IoT devices or to fuse physical and virtual technologies. More specifically, a new research area namely network slicing has emerged [257], which consists of dividing a single network connection into several virtual connections providing distinct portions of resources that serve different types of traffics. An interesting direction could be the enhancement of volumetric video streaming in 6G networks via resource slicing.

Another major transformation from previous cellular networks generations that 6G will bring, is the use of artificial intelligence (AI) in the entire system architecture, from core to the edge of the network. This transformation will be an evolution towards "connect intelligence". In the video streaming context, 6G networks can use AI to perform big data analytics of multimedia content to gain insights on network performance, channel conditions, viewers perspectives and preferences, and viewers' location prediction. MEC plays a key role in 6G networks as it can operate as an intermediate layer providing localized and low-latency data processing for real-time applications. More specifically, the high capacities of centralized cloud servers offer sophisticated video and metadata analysis, while sacrificing in terms of transmission overhead. Using the MEC allows to divide the data analytics into tow phases. First, fast content caching and processing can be done at the edge to perform critical tasks, owing to the proximity of MEC servers to viewers. As a second step, a deeper and more efficient data processing is carried out in the remote servers, at the expense of higher delays. Resource allocation decisions for live videos, cache placement of content, video recommendation, and advertising process can be cited as relevant use cases [255]. However, given the limited resources in MEC networks, computing the AI training/inferences in MEC devices may be infeasible, particularly when the task requires high computational load, e.g., Deep Neural Networks. A promising solution is to adopt pervasive computing, where different data storages and processing capacities existing everywhere and including distributed edge servers and users devices cooperate to accomplish AI tasks that need large memory and intensive computation. This marriage of pervasive computing in MEC networks and AI has given rise to a new research area, namely "Pervasive AI", which garnered considerable attention from both academia and industry [258]. Formally, pervasive AI focuses on how to intelligently distribute the inference or the training of the AI model across edge devices, to minimize the latency, and improve privacy and scalability. Research of this paradigm is still in its infancy and needs further investigation to improve the deployment of AI for video streaming tasks in mobile edge computing networks.

B. RESOURCE MIGRATION WITH USER MOBILITY

While the purpose of edge computing is to bring storage and computation closer to the user, mobile users often move around from one edge server to another one and resource migration is required to provide high quality of experience. Consider the example of a tourist moving around a city with interactive glasses through which he receives video content about the places he sees through it. While the video content is always served through the nearest edge server to the respective place he is currently viewing, the tourist might move away (e.g. sitting in a tourist bus) while he still wishes to continue watching the video of the previous place. In such a scenario, the previous edge server needs to migrate the content and computation to the new server closer to to the user's current location. Another use case when resource migration might be required across edge servers, is to emergency treatment provided to a critical patient in an ambulance moving towards the hospital. The ambulance on-board facilities are connected via 5G links to the healthcare provider central data and computing resources. As the ambulance is traveling, it changes association to different edge servers and consequently the resource migration is required. The resource migration in the above two use-cases have different latency constraints i.e., the second case have more stringent latency constraint. The current state-of-the-art in resource migration is not sufficient and more investigation is required to provide the required QoS in different use cases involving resource migration in edge networks.

C. LIVE VIDEO ANALYTICS FOR DRONES

Drones also termed as Unmanned Aerial Vehicles (UAVs) are becoming popular and affordable and their use in commercial application is growing. Drones are considered as a first choice for applications such as search-and-rescue, surveillance, and wildlife conservation. These applications usually require real-time video analytics which is a major challenge. Although the use of edge computing in drone-based video applications can bring significant improvement in latency and offloading intensive computation tasks, live video analytics is still the area which requires further research and state-of-the-art solutions. In particular these issues need further attention. (i) Drone's mobility may need computation migration from one edge server to another on the fly. (ii) The wireless links from drone to the edge server rapidly vary and the resource allocation shall be mandatory considering a joint communication and computation mechanism. (iii) Drones have limited onboard energy and only minimal computation to be performed locally to reduce energy consumption of drone. (iv) Drones that are receiving poor coverage by all edge servers may rely on D2D offloading to forward computation or caching access to other drones in the coverage. However, the effectiveness of such D2D cooperation is highly dependent upon the aerial links quality and stability. MEC-assisted drones [259] and drone-drone cooperation in video services involve large-sized content sharing and intensive offloading, which poses severe challenges as compared to normal drone cooperation.

D. MEC IN VR, AR, MR AND 360° VIDEOS STREAMING

Virtual Reality, Augmented Reality and Mixed Reality have been briefly discussed earlier in the paper. Recently 360° videos have been introduced, as a part of VR. 360° videos are large sized videos typically about 5x larger than that of conventional videos. Furthermore, the 360° videos require lower delivery latency and higher bandwidth compared to traditional video streaming. More importantly, the Quality of Experience (QoE) of viewers is highly sensitive to the dynamics of the network environment and it is extremely intolerant to delay variance and image freezing. Moreover, because of the huge increase of the number of VR devices which are estimated to reach 50 million in 2021 [260], the backhaul network becomes a bottleneck. As stated previously, the cloud wisdom is no longer sustainable for such real-time applications with stringent requirements. Therefore, MEC edge servers are introduced as an alternative to the cloud computing offering computation and caching capabilities in the vicinity of end-users. However, the huge processing requirements of 360° videos could quickly exhaust the available resources of the MEC Servers. Hence, efficient algorithms need to be designed for the given computational resources. Second, caching multiple views of video incurs high overhead in storage. Even if hard disks are not expensive nowadays, storing all of these files is neither economical nor feasible.

E. SECURE LIVE VIDEO STREAMING USING BLOCKCHAIN

To address the challenges of lives video streaming, the collaborative MEC servers process raw streams locally to serve viewers with minimal delays. However, the wireless data transmission incurs concerns in data security and privacy because it exposes vulnerabilities to potential attackers to perform malicious operations, such as eavesdropping private video streams (e.g. non-paying subscribers). To prevent the attackers from hacking the data, it is necessary to secure the communication channel between different edge nodes through encryption. However, due to the limited computational resources available at MEC servers and the time sensitivity of live streaming, the encryption will be relatively weak. Since the live streaming system is composed of a distributed network environment with a huge number of crowdsourcers with large heterogeneity, a dynamic, flexible, scalable and lightweight security mechanism is required. Additionally, the crowdsourcers (subscribers) are geographically distributed across an untrusted edge network. Hence, it is not suitable to implement security system on a centralized device, which may suffer from performance bottleneck. Blockchain [23] is a distributed database that contains chronologically connected data blocks. Each block is an individual component that includes information linked to a specific transaction. We envision that blockchain will be a strong fit to provide a suitable solution for secure live video sharing because of its immutability and decentralization features, which are perfectly consistent with our context. New requirements need to be handled in this task including, addressing the trade-off between increasing security level and latency. Public blockchain is slower than traditional database, since it coordinates the blocks of multiple unaffiliated participants, which contradicts the immediate responsiveness of live streaming. However, since we deal with already subscribed users of paid channels, blockchain solution can be feasible. A future direction could be the design of a complete architecture for blockchain-enabled authentication scheme for live videos systems in MEC networks and a comprehensive evaluation to validate the feasibility of the proposed authentication on the live streaming platforms.

IX. CONCLUSION

This paper surveys the research efforts on mobile edge computing for video streaming services. The paper presented a brief overview of mobile edge computing architecture and the various benefits it brings to users and network/content providers. Different schemes for edge-based content caching and processing are discussed and categorized. The potential benefits of mobile edge computing for video streaming services are presented. The state-of-the-art in MEC for video streaming is summarized. In addition to the MEC applications for video streaming services, D2D cooperation among end users to improve the edge computing performance is presented and research efforts in D2D-streaming are discussed. Machine learning techniques have been rigorously used in MEC systems and a significant amount of work has been done in this area. The paper summarised the use of ML/DL in MEC systems for video streaming applications. Lastly, some useful insights and future research directions are provided for prospective researchers.

ACKNOWLEDGMENT

The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] Netflix, Inc. *Netflix*. Accessed: Sep. 27, 2020. [Online]. Available: https://www.netflix.com/

[2] *Hulu*. Accessed: Sep. 27, 2020. [Online]. Available: http://www.hulu.com/

[3] Amazon, Inc. *Amazon Prime*. Accessed: Mar. 21, 2021. [Online]. Available: https://www.primevideo.com/

[4] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020.

[5] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," *J. Netw. Comput. Appl.*, vol. 169, Nov. 2020, Art. no. 102781.

[6] A. Shakarami, M. Ghobaei-Arani, M. Masdari, and M. Hosseinzadeh, "A survey on the computation offloading approaches in mobile edge/cloud computing environment: A stochastic-based perspective," *J. Grid Comput.*, vol. 18, no. 4, pp. 1–33, 2020.

[7] M. De Donno, K. Tange, and N. Dragoni, "Foundations and evolution of modern computing paradigms: Cloud, IoT, edge, and fog," *IEEE Access*, vol. 7, pp. 150936–150948, 2019.

[8] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Gener. Comput. Syst.*, vol. 97, pp. 219–235, Aug. 2019.

[9] N. Hassan, K.-L. A. Yau, and C. Wu, "Edge computing in 5G: A review," *IEEE Access*, vol. 7, pp. 127276–127289, 2019.

[10] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2017.

[11] 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.

[12] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge computing benefit from software-defined networking: A survey, use cases, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2359–2391, 4th Quart., 2017.

[13] E. Ahmed and M. H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," *Future Generat. Comput. Syst.*, vol. 70, pp. 59–63, May 2017.

[14] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.

[15] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," 2017, *arXiv:1701.01090*.

[16] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 1–8.

[17] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Jan. 2016.

[18] D. Prerna, R. Tekchandani, and N. Kumar, "Device-to-device content caching techniques in 5G: A taxonomy, solutions, and challenges," *Comput. Commun.*, vol. 153, pp. 48–84, Mar. 2020.

[19] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward computation offloading in edge computing: A survey," *IEEE Access*, vol. 7, pp. 131543–131558, 2019.

[20] L. Lin, X. Liao, H. Jin, and P. Li, "Computation offloading toward edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1584–1607, Aug. 2019.

[21] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[22] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, and W. Lv, "Edge computing security: State of the art and challenges," *Proc. IEEE*, vol. 107, no. 8, pp. 1608–1631, Aug. 2019.

[23] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508–1532, 2nd Quart., 2019.

[24] D. Liu, Z. Yan, W. Ding, and M. Atiquzzaman, "A survey on secure data analytics in edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4946–4967, Jun. 2019.

[25] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: Survey and open issues," *IEEE Access*, vol. 6, pp. 18209–18237, 2018.

[26] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, Jan. 2018.

[27] S. N. Shirazi, A. Gouglidis, A. Farshad, and D. Hutchison, "The extended cloud: Review and analysis of mobile edge computing and fog from a security and resilience perspective," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2586–2595, Nov. 2017.

[28] M. Marjanovic, A. Antonic, and I. P. Žarko, "Edge computing architecture for mobile crowdsensing," *IEEE Access*, vol. 6, pp. 10662–10674, 2018.

[29] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for Internet of Things realization," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2961–2991, 4th Quart., 2018.

[30] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2017.

[31] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.

[32] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7457–7469, Aug. 2020.

[33] G. Carvalho, B. Cabral, V. Pereira, and J. Bernardino, "Computation offloading in edge computing environments using artificial intelligence techniques," *Eng. Appl. Artif. Intell.*, vol. 95, Oct. 2020, Art. no. 103840.

[34] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.

[35] X. Jiang, F. R. Yu, T. Song, and V. C. M. Leung, "A survey on multi-access edge computing applied to video streaming: Some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 871–903, 2nd Quart., 2021.

[36] K. Kanai, K. Imagane, and J. Katto, "Overview of multimedia mobile edge computing," *ITE Trans. Media Technol. Appl.*, vol. 6, no. 1, pp. 46–52, 2018.

[37] Q. Zhang, H. Sun, X. Wu, and H. Zhong, "Edge video analytics for public safety: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1675–1696, Aug. 2019.

[38] Apple Inc. *iTunes (Home Box Office)*. Accessed: Sep. 27, 2020. [Online]. Available: https://www.apple.com/qa/itunes/

[39] *Cisco Annual Internet Report—Cisco Annual Internet Report (2018–2023) White Paper*, Cisco, San Jose, CA, USA, Mar. 2020.

[40] *League Legends*. Accessed: Sep. 27, 2020. [Online]. Available: https://na.leagueoflegends.com/en-us/

[41] *Counter-Strike: Global Offensive*. Accessed: Sep. 27, 2020. [Online]. Available: https://blog.counter-strike.net/

[42] *Overwatch*. Accessed: Sep. 27, 2020. [Online]. Available: https://playoverwatch.com/en-us/

[43] *Dota 2*. Accessed: Sep. 27, 2020. [Online]. Available: https://blog.dota2.com/

[44] *Call Duty*. Accessed: Sep. 27, 2020. [Online]. Available: https://blog.dota2.com/

[45] Twitter Inc. *Periscope*. Accessed: Sep. 27, 2020. [Online]. Available: https://www.pscp.tv/

[46] Google. *YouTube Live Streaming API Overview*. Accessed: Sep. 27, 2020. [Online]. Available: https://developers.google.com/youtube/v3/live/getting-started

[47] K. Pires and G. Simon, "YouTube live and twitch: A tour of user-generated live streaming systems," in *Proc. 6th ACM Multimedia Syst. Conf.*, New York, NY, USA, Mar. 2015, pp. 225–230.

[48] R. T. Sheffield, "Facebook live as a recordmaking technology," *Archivaria*, vol. 85, pp. 96–121, Jan. 2018.

[49] Instgram. *How do I View Someone's Live Video on Instagram?—Twitter Blog*. Accessed: Sep. 27, 2020. [Online]. Available: https://help.instagram.com/699289326902954

[50] Twitter. *Go Live on Twitter!—Twitter Blog*. Accessed: Sep. 27, 2020. [Online]. Available: https://blog.twitter.com/en_us/a/2016/go-live-on-twitter.html

[51] IBM. *IBM Cloud Video (Formerly Ustream)*. Accessed: Sep. 27, 2020. [Online]. Available: https://video.ibm.com/

[52] Dacast Inc. *Dacast*. Accessed: Sep. 27, 2020. [Online]. Available: https://www.dacast.com/

[53] *Kaltura*. Accessed: Sep. 27, 2020. [Online]. Available: https://corp.kaltura.com/

[54] *Vimeo Livestream*. Accessed: Sep. 27, 2020. [Online]. Available: https://vimeo.com/features/livestreaming

[55] *Panopto*. Accessed: Sep. 27, 2020. [Online]. Available: https://www.panopto.com/

[56] P. Dogga, S. Chakraborty, S. Mitra, and R. Netravali, "Edge-based transcoding for adaptive live video streaming," in *Proc. 2nd USENIX Workshop Hot Topics Edge Comput.*, 2019, pp. 1–7.

[57] M. Zink, R. Sitaraman, and K. Nahrstedt, "Scalable 360° video stream delivery: Challenges, solutions, and opportunities," *Proc. IEEE*, vol. 107, no. 4, pp. 639–650, Apr. 2019.

[58] J. Ruan and D. Xie, "A survey on QoE-oriented VR video streaming: Some research issues and challenges," *Electronics*, vol. 10, no. 17, p. 2155, Sep. 2021.

[59] Macromedia Inc. (2012). *Real-Time Messaging Protocol (RTMP) Specification*. [Online]. Available: https://www.adobe.com/devnet/rtmp.html

[60] H. Schulzrinne, A. Rao, R. Lanphier, M. Westerlund, and M. Stiemerling, *Real Time Streaming Protocol (RTSP) Version 2.0*, Standard RFC 7826, IETF, 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc7826

[61] R. Pantos. (2017). *HTTP Live Streaming*. Specification RFC 8216. [Online]. Available: https://www.rfc-editor.org/info/rfc8216

[62] Apple Inc. (2009). *Enabling Low-Latency HLS*. Accessed: Sep. 24, 2020. [Online]. Available: https://developer.apple.com/documentation/http_live_streaming/about_apple_s_http_live_streaming_tools

[63] *Information Technology—Dynamic Adaptive Streaming Over HTTP (Dash)—Part 1: Media Presentation Description and Segment Format*, Standard ISO/IEC 23009-1:2012, 2012. [Online]. Available: https://www.iso.org/standard/57623.html

[64] Apple Inc., *Common Media Application Format With HTTP Live Streaming*. Accessed: Sep. 24, 2020. [Online]. Available: https://developer.apple.com/documentation/http_live_streaming/about_the_common_media_application_format_with_http_live_streaming

[65] Microsoft Inc. (2009). *Microsof Smooth Streaming (MSS)*. Accessed: Sep. 24, 2020. [Online]. Available: https://docs.microsoft.com/en-us/iis/media/on-demand-smooth-streaming/smooth-streaming-technical-overview

[66] Adobe Systems Inc. (2009). *HTTP Dynamic Streaming*. Accessed: Sep. 24, 2020. [Online]. Available: https://www.adobe.com/products/hds-dynamic-streaming.html

[67] Open Source. *Secure Reliable Transport (SRT)*. Accessed: Sep. 24, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Secure_Reliable_Transport

[68] Open Source. *Real-Time Communication for the Web*. Accessed: Sep. 24, 2020. [Online]. Available: https://webrtc.org/

[69] WOWZA Media Systems. *2019 Video Streaming Latency Report*. Accessed: Sep. 27, 2020. [Online]. Available: https://www.wowza.com/blog/2019-video-streaming-latency-report

[70] N. Weil. *Part 1: How to Compete With Broadcast Latency Using Current Adaptive Bitrate Technologies*. Accessed: Sep. 24, 2020. [Online]. Available: https://aws.amazon.com/blogs/media/how-to-compete-with-broadcast-latency-using-current-adaptive-bitrate-technologies-part-1/

[71] *Parameter Values for Ultra-High Definition Television Systems for Production and International Programme Exchange*, International Telecommunication Union, Geneva, Switzerland, Oct. 2019.

[72] *Requirements for Mobile Edge Computing Enabled Content Delivery Networks*, International Telecommunication Union, Geneva, Switzerland, Mar. 2019.

[73] *Cisco Annual Internet Report (2018–2023) White Paper*, Cisco, San Jose, CA, USA, 2020.

[74] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.

[75] P. Yang, Y. Xiao, M. Xiao, and S. Li, "6G Wireless communications: Vision and potential techniques," *IEEE Netw.*, vol. 33, no. 4, pp. 70–75, Jul./Aug. 2019.

[76] N. Barber, "Prepare to support video livestreaming for customer experiences," Forrester Res., Cambridge, MA, USA, Tech. Rep. RES135382, 2017.

[77] *Thundering Herd Problem*. Accessed: Mar. 3, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Thundering_herd_problem

[78] Guardian. (2020). *HQ Trivia: The Gameshow App That's An Online Smash?*. Accessed: Aug. 8, 2020. [Online]. Available: https://www.theguardian.com/technology/2018/may/04/name-gameshow-app-thats-an-online-smash-hq-trivia

[79] *Exploding Watermelon Blows the Internet's Mind*. Accessed: Mar. 3, 2021. [Online]. Available: https://edition.cnn.com/2016/04/08/entertainment/buzzfeed-exploding-watermelon-irpt/index.html

[80] J. Greengrass, J. Evans, and A. C. Begen, "Not all packets are equal. Part I: Streaming video coding and SLA requirements," *IEEE Internet Comput.*, vol. 13, no. 1, pp. 70–75, Jan. 2009.

[81] E. Bertino, T. Catarci, A. K. Elmagarmid, and M. S. Hacid, "Quality of service specification in video databases," *IEEE MultimediaMag.*, vol. 10, no. 4, pp. 71–81, Oct. 2003.

[82] M. Venkataraman and M. Chatterjee, "Inferring video QoE in real time," *IEEE Netw.*, vol. 25, no. 1, pp. 4–13, Jan. 2011.

[83] *Quality of Experience Requirements for IPTV Services*, International Telecommunication Union, Geneva, Switzerland, Dec. 2008.

[84] T. Szigeti and C. Hattingh, "Quality of service design overview," in *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs*, J. Fagerberg, D. C. Mowery, and R. R. Nelson, Eds. San Jose, CA, USA: Cisco, 2004, ch. 2.

[85] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, and A. Neal, "Mobile-edge computing introductory technical white paper," Mobile-Edge Computing (MEC), Eur. Telecommun. Standards Inst., France, Tech. Rep. 1, pp. 1089–7801, 2014.

[86] H. S. Kim, "Fog computing and the Internet of Things: Extend the cloud to where the things are," Cisco, San Jose, CA, USA, Tech. Rep. 1, 2015.

[87] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," *Comput. Netw.*, vol. 130, pp. 94–120, Jan. 2018.

[88] F. Wang, M. Zhang, X. Wang, X. Ma, and J. Liu, "Deep learning for edge computing applications: A state-of-the-art survey," *IEEE Access*, vol. 8, pp. 58322–58336, 2020.

[89] A. Khreishah, I. Khalil, A. Gharaibeh, H. B. Salameh, and R. Alasem, "Joint caching and routing for greening computer networks with renewable energy sources," in *Proc. Int. Conf. Future Internet Things Cloud*, Aug. 2014, pp. 101–106.

[90] A. Khreishah, J. Chakareski, A. Gharaibeh, I. Khalil, and Y. Jararweh, "Joint data placement and flow control for cost-efficient data center networks," in *Proc. 6th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2015, pp. 274–279.

[91] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, "Collaborative multi-bitrate video caching and processing in mobile-edge computing networks," in *Proc. 13th Annu. Conf. Wireless Demand Netw. Syst. Services (WONS)*, Feb. 2017, pp. 165–172.

[92] C. Ge, N. Wang, G. Foster, and M. Wilson, "Toward QoE-assured 4K video-on-demand delivery through mobile edge virtualization with adaptive prefetching," *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2222–2237, Oct. 2017.

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

[94] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[95] Y. Hao, L. Hu, Y. Qian, and M. Chen, "Profit maximization for video caching and processing in edge cloud," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1632–1641, Jul. 2019.

[96] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[97] H. Wu, Q. Wang, and K. Wolter, "Tradeoff between performance improvement and energy saving in mobile cloud offloading systems," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2013, pp. 728–732.

[98] F. Wei, S. Chen, and W. Zou, "A greedy algorithm for task offloading in mobile edge computing system," *China Commun.*, vol. 15, no. 11, pp. 149–157, Nov. 2018.

[99] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[100] M. Jia, J. Cao, and L. Yang, "Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2014, pp. 352–357.

[101] L. Yang, J. Cao, S. Tang, T. Li, and A. T. S. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," in *Proc. IEEE 5th Int. Conf. Cloud Comput.*, Jun. 2012, pp. 794–802.

[102] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

[103] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.

[104] B. Dab, N. Aitsaadi, and R. Langar, "Joint optimization of offloading and resource allocation scheme for mobile edge computing," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–7.

[105] T. Yang, Y. Hu, M. C. Gursoy, A. Schmeink, and R. Mathar, "Deep reinforcement learning based resource allocation in low latency edge computing networks," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2018, pp. 1–5.

[106] F. Haouari, E. Baccour, A. Erbad, A. Mohamed, and M. Guizani, "Transcoding resources forecasting and reservation for crowdsourced live streaming," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–7.

[107] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Joint allocation of computation and communication resources in multiuser mobile cloud computing," in *Proc. IEEE 14th Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2013, pp. 26–30.

[108] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5031–5044, May 2019.

[109] M. Molina, O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Joint scheduling of communication and computation resources in multiuser wireless application offloading," in *Proc. IEEE 25th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2014, pp. 1093–1098.

[110] S. Guo, B. Xiao, Y. Yang, and Y. Yang, "Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[111] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski, "Edge computing assisted adaptive mobile video streaming," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 787–800, Jun. 2019.

[112] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3887–3901, Dec. 2016.

[113] D. Wu, F. Wang, X. Cao, and J. Xu, "Joint communication and computation optimization for wireless powered mobile edge computing with D2D offloading," *J. Commun. Inf. Netw.*, vol. 4, no. 4, pp. 72–86, Dec. 2019.

[114] P. Yuan, Y. Cai, X. Huang, S. Tang, and X. Zhao, "Collaboration improves the capacity of mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10610–10619, Dec. 2019.

[115] H. Xing, L. Liu, J. Xu, and A. Nallanathan, "Joint task assignment and resource allocation for D2D-enabled mobile-edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4193–4207, Jun. 2019.

[116] F. Haouari, E. Baccour, A. Erbad, A. Mohamed, and M. Guizani, "QoE-aware resource allocation for crowdsourced live streaming: A machine learning approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[117] X. Niu, S. Shoa, C. Xiu, J. Zhou, S. Guo, X. Chen, and F. Qi, "Workload allocation mechanism for minimum service delay in edge computing-based power Internet of Things," *IEEE Access*, vol. 7, pp. 83771–83784, 2019.

[118] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–6.

[119] K. Bilal, E. Baccour, A. Erbad, A. Mohamed, and M. Guizani, "Collaborative joint caching and transcoding in mobile edge networks," *J. Netw. Comput. Appl.*, vol. 136, pp. 86–99, Jun. 2019.

[120] S. Wang, R. Urgaonkar, T. He, M. Zafer, K. Chan, and K. K. Leung, "Mobility-induced service migration in mobile micro-clouds," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2014, pp. 835–840.

[121] R. Urgaonkar, S. Wang, T. He, M. Zafer, K. Chan, and K. K. Leung, "Dynamic service migration and workload scheduling in edge-clouds," *Perform. Eval.*, vol. 91, pp. 205–228, Sep. 2015.

[122] M.-H. Chen, B. Liang, and M. Dong, "Joint offloading decision and resource allocation for multi-user multi-task mobile cloud," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[123] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129–140, Feb. 2013.

[124] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1357–1370, Oct. 2009.

[125] H. Ahlehagh and S. Dey, "Video-aware scheduling and caching in the radio access network," *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1444–1462, Oct. 2014.

[126] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

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

[128] Platform9 Blog. *Edge Computing and Video Streaming: Improving User Experience*. Accessed: Nov. 4, 2020. [Online]. Available: https://platform9.com/blog/edge-computing-and-video-streaming-improving-user-experience/

[129] (2020). *Real-Life Use Cases for Edge Computing*. Accessed: Nov. 4, 2020. [Online]. Available: https://innovationatwork.ieee.org/real-life-edge-computing-use-cases/

[130] K. Bilal and A. Erbad, "Edge computing for interactive media and video streaming," in *Proc. 2nd Int. Conf. Fog Mobile Edge Comput. (FMEC)*, May 2017, pp. 68–73.

[131] S.-R. Yang, Y.-J. Tseng, C.-C. Huang, and W.-C. Lin, "Multi-access edge computing enhanced video streaming: Proof-of-concept implementation and prediction/QoE models," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1888–1902, Feb. 2019.

[132] H. Hu, H. Shan, C. Wang, T. Sun, X. Zhen, K. Yang, L. Yu, Z. Zhang, and T. Q. S. Quek, "Video surveillance on mobile edge networks—A reinforcement-learning-based approach," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4746–4760, Jun. 2020.

[133] H. Hu, H. Shan, Z. Zheng, Z. Huang, C. Cai, C. Wang, X. Zhen, L. Yu, Z. Zhang, and T. Q. S. Quek, "Intelligent video surveillance based on mobile edge networks," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*, Dec. 2018, pp. 286–291.

[134] Z. Rao and Z. Guo, "Application of video analysis based on mobile edge computing," in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, Dec. 2017, pp. 2040–2044.

[135] Y. Zhang, J.-H. Liu, C.-Y. Wang, and H.-Y. Wei, "Decomposable intelligence on cloud-edge IoT framework for live video analytics," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8860–8873, Sep. 2020.

[136] X. Jiang, F. R. Yu, T. Song, and V. C. M. Leung, "Intelligent resource allocation for video analytics in blockchain-enabled Internet of Autonomous vehicles with edge computing," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14260–14272, Aug. 2022.

[137] H. Sun, Y. Yu, K. Sha, and B. Lou, "MVideo: Edge computing based mobile video processing systems," *IEEE Access*, vol. 8, pp. 11615–11623, 2020.

[138] E. Blog. (2020). *How 5G and Edge Computing Can Enhance Virtual Reality*. Accessed: Nov. 4, 2020. [Online]. Available: https://www.ericsson.com/en/blog/2020/4/how-5g-and-edge-computing-can-enhance-virtual-reality

[139] Multimodal Educatyion Center. (2020). *Virtual Reality Experiences at the MEC*. Accessed: Nov. 4, 2020. [Online]. Available: https://www.cwu.edu/multimodal-education/virtual-reality-experiences-mec

[140] (2021). *TriMirror*. [Online]. Available: https://www.trimirror.com/Solutions/InStore

[141] C. Shearer, D. Paul, B. S. William, and L. Hazlewood, ''Blended reality systems and methods,'' U.S. Patent 9 858 719, Jan. 2018.

[142] S. Sukhmani, M. Sadeghi, M. Erol-Kantarci, and A. El Saddik, ''Edge caching and computing in 5G for mobile AR/VR and tactile internet,'' *IEEE Multimedia Mag.*, vol. 26, no. 1, pp. 21–30, Jan./Mar. 2019.

[143] J. Dai and D. Liu, ''An MEC-enabled wireless VR transmission system with view synthesis-based caching,'' in *Proc. IEEE Wireless Commun. Netw. Conf. Workshop (WCNCW)*, Apr. 2019, pp. 1–7.

[144] J. Dai, Z. Zhang, S. Mao, and D. Liu, ''A view synthesis-based 360° VR caching system over MEC-enabled C-RAN,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3843–3855, Oct. 2020.

[145] (2020). *YouTube at 15: My Personal Journey and the Road Ahead*. [Online]. Available: https://blog.youtube/news-and-events/youtube-at-15-my-personal-journey

[146] D. K. Krishnappa, S. Khemmarat, L. Gao, and M. Zink, ''On the feasibility of prefetching and caching for online TV services: A measurement study on hulu,'' in *Passive and Active Measurement*, N. Spring and G. F. Riley, Eds. Berlin, Germany: Springer, 2011, pp. 72–80.

[147] S. Li, J. Xu, M. van der Schaar, and W. Li, ''Popularity-driven content caching,'' in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[148] 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.

[149] H. Li, X. Ma, F. Wang, J. Liu, and K. Xu, ''On popularity prediction of videos shared in online social networks,'' in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 169–178.

[150] Y. Zhou, L. Chen, C. Yang, and D. M. Chiu, ''Video popularity dynamics and its implication for replication,'' *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1273–1285, Aug. 2015.

[151] D. Lee, J. Choi, J.-H. Kim, S. H. Noh, S. L. Min, Y. Cho, and C. S. Kim, ''LRFU: A spectrum of policies that subsumes the least recently used and least frequently used policies,'' *IEEE Trans. Comput.*, vol. 50, no. 12, pp. 1352–1361, Dec. 2001.

[152] Y. Zeng, J. Xie, H. Jiang, G. Huang, S. Yi, N. Xiong, and J. Li, ''Smart caching based on user behavior for mobile edge computing,'' *Inf. Sci.*, vol. 503, pp. 444–468, Nov. 2019.

[153] 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. 10, pp. 1863–1876, Aug. 2016.

[154] E. Baccour, A. Erbad, A. Mohamed, K. Bilal, and M. Guizani, ''Proactive video chunks caching and processing for latency and cost minimization in edge networks,'' in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–7.

[155] K. N. Doan, T. Van Nguyen, T. Q. S. Quek, and H. Shin, ''Content-aware proactive caching for backhaul offloading in cellular network,'' *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3128–3140, May 2018.

[156] S. He, H. Tian, and X. Lyu, ''Edge popularity prediction based on social-driven propagation dynamics,'' *IEEE Commun. Lett.*, vol. 21, no. 5, pp. 1027–1030, May 2017.

[157] A.-T. Tran, N.-N. Dao, and S. Cho, ''Bitrate adaptation for video streaming services in edge caching systems,'' *IEEE Access*, vol. 8, pp. 135844–135852, 2020.

[158] Z. Qu, B. Ye, B. Tang, S. Guo, S. Lu, and W. Zhuang, ''Cooperative caching for multiple bitrate videos in small cell edges,'' *IEEE Trans. Mobile Comput.*, vol. 19, no. 2, pp. 288–299, Feb. 2020.

[159] Z. Su, Q. Xu, F. Hou, Q. Yang, and Q. Qi, ''Edge caching for layered video contents in mobile social networks,'' *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2210–2221, Oct. 2017.

[160] L. Vigneri, T. Spyropoulos, and C. Barakat, ''Low cost video streaming through mobile edge caching: Modelling and optimization,'' *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1302–1315, Jun. 2019.

[161] C. Zhang, H. Pang, L. Gao, Q. Ding, Y. Yang, and L. Sun, ''Incentivizing mobile video users with data sponsoring and edge caching,'' *IEEE Access*, vol. 8, pp. 9640–9654, 2020.

[162] Y. Cheng, ''Edge caching and computing in 5G for mobile augmented reality and haptic internet,'' *Comput. Commun.*, vol. 158, pp. 24–31, May 2020.

[163] S. Kumar, S. V. Doddala, A. A. Franklin, and J. Jin, ''RAN-aware adaptive video caching in multi-access edge computing networks,'' *J. Netw. Comput. Appl.*, vol. 168, Oct. 2020, Art. no. 102737.

[164] L. Li, D. Shi, R. Hou, R. Chen, B. Lin, and M. Pan, ''Energy-efficient proactive caching for adaptive video streaming via data-driven optimization,'' *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5549–5561, Jun. 2020.

[165] Z. Yan, M. Zhao, C. Westphal, and C. W. Chen, ''Toward guaranteed video experience: Service-aware downlink resource allocation in mobile edge networks,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 6, pp. 1819–1831, Jun. 2019.

[166] X. Yang, Z. Chen, K. Li, Y. Sun, N. Liu, W. Xie, and Y. Zhao, ''Communication-constrained mobile edge computing systems for wireless virtual reality: Scheduling and tradeoff,'' *IEEE Access*, vol. 6, pp. 16665–16677, 2018.

[167] T. X. Tran and D. Pompili, ''Adaptive bitrate video caching and processing in mobile-edge computing networks,'' *IEEE Trans. Mobile Comput.*, vol. 18, no. 9, pp. 1965–1978, Sep. 2019.

[168] Q. Tang, R. Xie, T. Huang, and Y. Liu, ''Jointly caching and computation resource allocation for mobile edge networks,'' *IET Netw.*, vol. 8, no. 5, pp. 329–338, Sep. 2019.

[169] T. Zhang and S. Mao, ''Joint video caching and processing for multi-bitrate videos in ultra-dense HetNets,'' *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1230–1243, 2020.

[170] E. Baccour, A. Erbad, K. Bilal, A. Mohamed, and M. Guizani, ''PCCP: Proactive video chunks caching and processing in edge networks,'' *Future Gener. Comput. Syst.*, vol. 105, pp. 44–60, Apr. 2020.

[171] N. Zhang, S. Guo, Y. Dong, and D. Liu, ''Joint task offloading and data caching in mobile edge computing networks,'' *Comput. Netw.*, vol. 182, Dec. 2020, Art. no. 107446.

[172] T. Dang and M. Peng, ''Joint radio communication, caching, and computing design for mobile virtual reality delivery in fog radio access networks,'' *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1594–1607, Jul. 2019.

[173] Y. Sun, Z. Chen, M. Tao, and H. Liu, ''Communications, caching, and computing for mobile virtual reality: Modeling and tradeoff,'' *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7573–7586, Nov. 2019.

[174] J. Luo, F. R. Yu, Q. Chen, L. Tang, and Z. Zhang, ''Adaptive video streaming in software-defined mobile networks: A deep reinforcement learning approach,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1577–1592.

[175] C. Wang, D. Feng, S. Zhang, and Q. Chen, ''Video caching and transcoding in wireless cellular networks with mobile edge computing: A robust approach,'' *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9234–9238, Aug. 2020.

[176] S. Rezvani, S. Parsaeefard, N. Mokari, M. R. Javan, and H. Yanikomeroglu, ''Delivery-aware cooperative joint multi-bitrate video caching and transcoding in 5G,'' *arXiv*, vol. abs/1805.07132, 2018.

[177] R. Yang and S. Guo, ''A mobile edge caching strategy for video grouping in vehicular networks,'' in *Proc. 13th Int. Conf. Adv. Comput. Intell. (ICACI)*, May 2021, pp. 40–45.

[178] 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.

[179] Y. Wang, Y. Zhang, M. Sheng, and K. Guo, ''On the interaction of video caching and retrieving in multi-server mobile-edge computing systems,'' *IEEE Wireless Commun. Lett.*, vol. 8, no. 5, pp. 1444–1447, Oct. 2019.

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

[181] Z. Sang, S. Guo, and Y. Wang, ''Collaborative video cache management strategy in mobile edge computing,'' in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2021, pp. 1–6.

[182] A.-E.-M. Taha, N. Abu Ali, H. R. Chi, and A. Radwan, ''MEC resource offloading for QoE-aware HAS video streaming,'' in *Proc. IEEE Int. Conf. Commun.*, Jun. 2021, pp. 1–5.

[183] R. Xie, Z. Li, J. Wu, Q. Jia, and T. Huang, ''Energy-efficient joint caching and transcoding for HTTP adaptive streaming in 5G networks with mobile edge computing,'' *China Commun.*, vol. 16, no. 7, pp. 229–244, Jul. 2019.

[184] X. Huang, L. He, L. Wang, and F. Li, ''Towards 5G: Joint optimization of video segment caching, transcoding and resource allocation for adaptive video streaming in a multi-access edge computing network,'' *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10909–10924, Oct. 2021.

[185] Q. I. Cheng, H. Shan, W. Zhuang, L. Yu, Z. Zhang, and T. Q. S. Quek, "Design and analysis of MEC- and proactive caching-based 360° mobile VR video streaming," *IEEE Trans. Multimedia*, vol. 24, pp. 1529–1544, 2021.

[186] C. Liu, H. Zhang, H. Ji, and X. Li, "MEC-assisted flexible transcoding strategy for adaptive bitrate video streaming in small cell networks," *China Commun.*, vol. 18, no. 2, pp. 200–214, Feb. 2021.

[187] E. Yaacoub, Z. Dawy, and A. Abu-Dayya, "On real-time video streaming over LTE networks with mobile-to-mobile cooperation," in *Proc. 19th Int. Conf. Telecommun. (ICT)*, Apr. 2012, pp. 1–6.

[188] M. A. Khan, R. Hamila, M. S. Kiranyaz, and M. Gabbouj, "A novel UAV-aided network architecture using Wi-Fi direct," *IEEE Access*, vol. 7, pp. 67305–67318, 2019.

[189] M. A. Khan, R. Hamila, and M. O. Hasna, "Optimal group formation in dense Wi-Fi direct networks for content distribution," *IEEE Access*, vol. 7, pp. 161231–161245, 2019.

[190] Y. Li, L. Sun, and W. Wang, "Exploring device-to-device communication for mobile cloud computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 2239–2244.

[191] W. Hu and G. Cao, "Quality-aware traffic offloading in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3182–3195, Nov. 2017.

[192] Y. He, J. Ren, G. Yu, and Y. Cai, "D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks," *IEEE Trans. Commun.*, vol. 18, no. 3, pp. 1750–1763, Mar. 2019.

[193] G. Qiao, S. Leng, and Y. Zhang, "Online learning and optimization for computation offloading in D2D edge computing and networks," *Mobile Netw. Appl.*, vol. 27, no. 3, pp. 1111–1122, Jun. 2022.

[194] P. Gope, J. Lee, R.-H. Hsu, and T. Q. S. Quek, "Anonymous communications for secure device-to-device-aided fog computing: Architecture, challenges, and solutions," *IEEE Consum. Electron. Mag.*, vol. 8, no. 3, pp. 10–16, May 2019.

[195] G. S. Park and H. Song, "Video quality-aware traffic offloading system for video streaming services over 5G networks with dual connectivity," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5928–5943, Jun. 2019.

[196] X. Zhang, H. Lin, M. Chen, B. Kang, and L. Wang, "MEC-enabled video streaming in device-to-device networks," *IET Commun.*, vol. 14, no. 15, pp. 2453–2461, Sep. 2020.

[197] J. Kim, T. Kim, M. Hashemi, C. G. Brinton, and D. J. Love, "Joint optimization of signal design and resource allocation in wireless D2D edge computing," 2020, *arXiv:2002.11850*.

[198] J. Kim, T. Kim, M. Hashemi, C. G. Brinton, and D. J. Love, "Joint optimization of signal design and resource allocation in wireless D2D edge computing," in *Proc. IEEE Conf. Comput. Commun.*, Jul. 2020, pp. 2086–2095.

[199] Y. Kim, F. Baccelli, and G. de Veciana, "Spatial reuse and fairness of ad hoc networks with channel-aware CSMA protocols," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 4139–4157, Jul. 2014.

[200] N. Naderializadeh and A. S. Avestimehr, "ITLinQ: A new approach for spectrum sharing in device-to-device communication systems," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1139–1151, Jun. 2014.

[201] S. Goebbels and R. Jennen, "Enhancements in wireless broadband networks using smart caching an analytical evaluation," in *Proc. IEEE 19th Int. Symp. Pers., Indoor Mobile Radio Commun.*, Sep. 2008, pp. 1–5.

[202] M. Dehghan, A. Seetharam, B. Jiang, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman, "On the complexity of optimal routing and content caching in heterogeneous networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 936–944.

[203] 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.

[204] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3923–3949, Jun. 2017.

[205] X. Wu, S. Tavildar, S. Shakkottai, T. Richardson, J. Li, R. Laroia, and A. Jovicic, "FlashLinQ: A synchronous distributed scheduler for peer-to-peer ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1215–1228, Aug. 2013.

[206] U. Abbasi and H. Elbiaze, "Multimedia streaming using D2D in 5G ultra dense networks," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2018, pp. 1–6.

[207] Q. Ren, J. Chen, B. Chen, and L. Jin, "A video streaming transmission scheme based on frame priority in Device-to-Device multicast networks," *IEEE Access*, vol. 7, pp. 20187–20198, 2019.

[208] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[209] Z. Chkirbene, A. Mohamed, A. Erbad, and M. Guizani, "Smart edge healthcare data sharing system," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2020, pp. 577–582.

[210] Z. Chkirbene, A. Erbad, R. Hamila, A. Gouissem, A. Mohamed, M. Guizani, and M. Hamdi, "A weighted machine learning-based attacks classification to alleviating class imbalance," *IEEE Syst. J.*, vol. 15, no. 4, pp. 4780–4791, Dec. 2021.

[211] A. Redondi, L. Baroffio, L. Bianchi, M. Cesana, and M. Tagliasacchi, "Compress-then-analyze versus analyze-then-compress: What is best in visual sensor networks?" *IEEE Trans. Mobile Comput.*, vol. 15, no. 12, pp. 3000–3013, Dec. 2016.

[212] C. W. Chen, "Internet of Video Things: Next-generation IoT with visual sensors," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6676–6685, Aug. 2020.

[213] Y. Cao, Z. Xu, P. Qin, and T. Jiang, "Video processing on the edge for multimedia IoT systems," 2018, *arXiv:1805.04837*.

[214] I. E. Olatunji and C.-H. Cheng, "Dynamic threshold for resource tracking in observed scenes," in *Proc. 9th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Jul. 2018, pp. 1–6.

[215] A. Alam, I. Ullah, and Y.-K. Lee, "Video big data analytics in the cloud: A reference architecture, survey, opportunities, and open research issues," *IEEE Access*, vol. 8, pp. 152377–152422, 2020.

[216] B. Marr, *Big Data: Using SMART Big Data, Analytics and Metrics to Make Better Decisions and Improve Performance*. Hoboken, NJ, USA: Wiley, 2015.

[217] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. Conf. ACM Special Interest Group Data Commun.*, 2017, pp. 197–210.

[218] T. Huang, R.-X. Zhang, C. Zhou, and L. Sun, "QARC: Video quality aware rate control for real-time video streaming based on deep reinforcement learning," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 1208–1216.

[219] G. Xiao, M. Wu, Q. Shi, Z. Zhou, and X. Chen, "DeepVR: Deep reinforcement learning for predictive panoramic video streaming," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 4, pp. 1167–1177, Dec. 2019.

[220] 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.

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

[222] 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.

[223] C. Zhang, H. Pang, J. Liu, S. Tang, R. Zhang, D. Wang, and L. Sun, "Toward edge-assisted video content intelligent caching with long short-term memory learning," *IEEE Access*, vol. 7, pp. 152832–152846, 2019.

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

[225] L. Huang, X. Feng, C. Zhang, L. Qian, and Y. Wu, "Deep reinforcement learning-based joint task offloading and bandwidth allocation for multi-user mobile edge computing," *Digit. Commun. Netw.*, vol. 5, no. 1, pp. 10–17, 2018.

[226] M. De Filippo De Grazia, D. Zucchetto, A. Testolin, A. Zanella, M. Zorzi, and M. Zorzi, "QoE multi-stage machine learning for dynamic video streaming," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 1, pp. 146–161, Mar. 2018.

[227] F. Wang, C. Zhang, F. Wang, J. Liu, Y. Zhu, H. Pang, and L. Sun, "Intelligent edge-assisted crowdcast with deep reinforcement learning for personalized QoE," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 910–918.

[228] W. Yang, X. Chi, L. Zhao, and Z. Xiong, "QoE-based MEC-assisted predictive adaptive video streaming for on-road driving scenarios," *IEEE Wireless Commun. Lett.*, vol. 10, no. 11, pp. 2552–2556, Nov. 2021.

[229] J. Ren, Y. Guo, D. Zhang, Q. Liu, and Y. Zhang, "Distributed and efficient object detection in edge computing: Challenges and solutions," *IEEE Netw.*, vol. 32, no. 6, pp. 137–143, Nov./Dec. 2018.

[230] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, M. Yunsheng, S. Chen, and P. Hou, "A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure," *IEEE Trans. Services Comput.*, vol. 11, no. 2, pp. 249–261, Mar. 2017.

[231] X. Ran, H. Chen, X. Zhu, Z. Liu, and J. Chen, "DeepDecision: A mobile deep learning framework for edge video analytics," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2018, pp. 1421–1429.

[232] D. Li, T. Salonidis, N. V. Desai, and M. C. Chuah, "DeepCham: Collaborative edge-mediated adaptive deep learning for mobile object recognition," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 64–76.

[233] S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B.-Y. Choi, and T. Faughnan, "Smart surveillance as an edge network service: From Harr-cascade, SVM to a lightweight CNN," in *Proc. IEEE 4th Int. Conf. Collaboration Internet Comput. (CIC)*, Oct. 2018, pp. 256–265.

[234] Y. Wang, Y. Dong, S. Guo, Y. Yang, and X. Liao, "Latency-aware adaptive video summarization for mobile edge clouds," *IEEE Trans. Multimedia*, vol. 22, no. 5, pp. 1193–1207, May 2020.

[235] A. Galanopoulos, J. A. Ayala-Romero, D. J. Leith, and G. Iosifidis, "AutoML for video analytics with edge computing," in *Proc. IEEE Conf. Comput. Commun.*, May 2021, pp. 1–10.

[236] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, Jan. 2016.

[237] E. Baccour, A. Erbad, A. Mohamed, M. Guizani, and M. Hamdi, "CE-D2D: Collaborative and popularity-aware proactive chunks caching in edge networks," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2020, pp. 1770–1776.

[238] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[239] X. Hou, S. Dey, J. Zhang, and M. Budagavi, "Predictive view generation to enable mobile 360-degree and VR experiences," in *Proc. Morning Workshop Virtual Reality Augmented Reality Netw.*, Aug. 2018, pp. 20–26.

[240] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, "Gaze prediction in dynamic 360 immersive videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5333–5342.

[241] S. Afzal, J. Chen, and K. K. Ramakrishnan, "Characterization of 360-degree videos," in *Proc. Workshop Virtual Reality Augmented Reality Netw.*, Aug. 2017, pp. 1–6.

[242] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proc. 12th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2014, pp. 68–81.

[243] E. Baccour, A. Erbad, K. Bilal, A. Mohamed, M. Guizani, and M. Hamdi, "FacebookVideoLive18: A live video streaming dataset for streams metadata and online viewers locations," in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Feb. 2020, pp. 476–483.

[244] Google Research. *YouTube-8M Segments Dataset*. Accessed: Nov. 3, 2020. [Online]. Available: http://research.google.com/youtube8m/

[245] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A dataset for exploring user behaviors in VR spherical video streaming," in *Proc. 8th ACM Multimedia Syst. Conf.*, Taipei, Taiwan, Jun. 2017, pp. 193–198.

[246] A. T. Nasrabadi, A. Samiei, A. Mahzari, R. P. Mcmahan, R. Prakash, M. C. Q. Farias, and M. M. Carvalho, "A taxonomy and dataset for 360° videos," in *Proc. 10th ACM Multimedia Syst. Conf.*, New York, NY, USA, Jun. 2019, pp. 273–278.

[247] D. Ghadiyaram, J. Pan, and A. C. Bovik, "A subjective and objective study of stalling events in mobile streaming videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 183–197, Jan. 2019.

[248] F. Loh, F. Wamser, C. Moldovan, B. Zeidler, T. Hoßfeld, D. Tsilimantos, and S. Valentin, "From click to playback: A dataset to study the response time of mobile YouTube," in *Proc. 10th ACM Multimedia Syst. Conf.*, New York, NY, USA, Jun. 2019, pp. 267–272.

[249] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: A 5G dataset with channel and context metrics," in *Proc. 11th ACM Multimedia Syst. Conf.*, New York, NY, USA, May 2020, pp. 303–308.

[250] A. Mercat, M. Viitanen, and J. Vanne, "UVG dataset: 50/120 fps 4K sequences for video codec analysis and development," in *Proc. 11th ACM Multimedia Syst. Conf.*, New York, NY, USA, May 2020, pp. 297–302.

[251] C. G. Bampis, Z. Li, I. Katsavounidis, T.-Y. Huang, C. Ekanadham, and A. Bovik, "Towards perceptually optimized end-to-end adaptive video streaming," 2018, *arXiv:1808.03898*.

[252] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, 2019, pp. 1–16.

[253] K. Boos, D. Chu, and E. Cuervo, "Flashback: Immersive virtual reality on mobile devices via rendering memoization," in *Proc. MobiSys*, 2016, pp. 291–304.

[254] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless communications and applications above 100 GHz: Opportunities and challenges for 6G and beyond," *IEEE Access*, vol. 7, pp. 78729–78757, 2019.

[255] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, Aug. 2019.

[256] A. Pennington. (2020). *5G's Future is Broken. Here's Why we Need 6G*. Accessed: Sep. 14, 2021. [Online]. Available: https://www.streamingmedia.com/Articles/Editorial/Featured-Articles/5Gs-Future-Is-Broken.-Heres-Why-We-Need-6G-143474.aspx?utm_source=related_articles&utm_medium=gutenberg&utm_campaign=editors_selection

[257] S. Wijethilaka and M. Liyanage, "Survey on network slicing for Internet of Things realization in 5G networks," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 957–994, 2nd Quart., 2021.

[258] E. Baccour, N. Mhaisen, A. A. Abdellatif, A. Erbad, A. Mohamed, M. Hamdi, and M. Guizani, "Pervasive AI for IoT applications: Resource-efficient distributed artificial intelligence," 2021, *arXiv:2105.01798*.

[259] J. Wang, Z. Feng, Z. Chen, S. A. George, M. Bala, P. Pillai, S.-W. Yang, and M. Satyanarayanan, "Edge-based live video analytics for drones," *IEEE Internet Comput.*, vol. 23, no. 4, pp. 27–34, Jul. 2019.

[260] C. V. Forecast, "Cisco visual networking index: Global mobile data traffic forecast update," Cisco, San Jose, CA, USA, Tech. Rep. c11-741490, 2017.

MUHAMMAD ASIF KHAN (Senior Member, IEEE) received the B.Sc. degree in telecommunication engineering from the University of Engineering and Technology Peshawar, Pakistan, in 2009, the M.Sc. degree in telecommunication engineering from the University of Engineering and Technology Taxila, Pakistan, in 2013, and the Ph.D. degree in electrical engineering from Qatar University, in 2020. He was a Research Assistant at Qatar University, from 2014 to 2015, and Qatar Mobility Innovation Center, from 2016 to 2017. He is currently working as a Postdoctoral Researcher at Qatar University. His current research interests include wireless communication, mobile edge computing, machine learning, and distributed optimization. He is a member of IET. For more information visit the link (http://www.asifk.me).

EMNA BACCOUR received the Ph.D. degree in computer Science from the University of Burgundy, France, in 2017. She was a Postdoctoral Fellow at Qatar University on a project covering the interconnection networks for massive data centers and then on a project covering video caching and processing in mobile edge computing networks. She currently holds a postdoctoral position at Hamad Ben Khalifa University. Her research interests include data center networks, cloud computing, green computing, software defined networks, and distributed systems. She is also interested in edge networks, mobile edge caching and computing, and the IoT systems.

ZINA CHKIRBENE received the bachelor's degree in computer science networks and telecommunications from the National Institute of Applied Science and Technology, in 2011, the master's degree in electronic systems and communication networks from the Polytechnic School, in 2012, and the Ph.D. degree in computer science from the University of Burgundy Dijon, France, in 2017. She was a Research Assistant at Qatar University on a project covering the interconnection networks for massive data centers, where she currently holds a postdoctoral position. Her research interests include data center networks, edge computing, green computing, machine learning, and deep reinforcement learning techniques.

AIMAN ERBAD (Senior Member, IEEE) received the M.Sc. degree from the University of Essex, in 2005, and the Ph.D. degree from The University of British Columbia, in 2012. He is currently an Associate Professor with the College of Science and Engineering, Hamad Bin Khalifa University (HBKU). His research interests include cloud computing, edge computing, the IoT, private and secure networks, and multimedia systems. He received the Platinum Award from H. H. Emir Sheikh Tamim bin Hamad Al Thani at the Education Excellence Day 2013 (Ph.D. category). He also received the 2020 Best Research Paper Award from Computer Communications, the IWCMC 2019 Best Paper Award, and the IEEE CCWC 2017 Best Paper Award. His research interests include cloud computing, edge computing, the IoT, distributed AI algorithms, and private/secure networks. He is currently an Editor of the *International Journal of Sensor Networks* (IJSNet) and *KSII Transactions on Internet and Information Systems*, and served as a Guest Editor for *IEEE Networks*.

RIDHA HAMILA (Senior Member, IEEE) received the M.Sc., Lic.Tech. (Hons.), and Ph.D. degrees from Tampere University, Tampere, Finland, in 1996, 1999, and 2002, respectively. He is currently a Full Professor at the Department of Electrical Engineering, Qatar University, Qatar. From 1994 to 2002, he held various research and teaching positions at the Department of Information Technology, TUT, Finland. From 2002 to 2003, he was a System Specialist at the Nokia research Center and Nokia Networks, Helsinki. From 2004 to 2009, he was with Emirates Telecommunications Corporation, United Arab Emirates. From 2004 to 2013, he was an Adjunct Professor at the Department of Communications Engineering, TUT. His current research interests include mobile and broadband wireless communication systems, mobile edge computing, the Internet of Everything, and machine learning. In these areas, he has published over 200 journals and conference papers most of them in the peer-reviewed IEEE publications, filed seven U.S. patents, and wrote numerous confidential industrial research reports. He has been involved in several past and current industrial projects, Ooredoo, Qatar National Research Fund, Finnish Academy projects, and EU research and education programs. He has supervised a large number of bachelor's students and postdoctoral fellows. He organized many international workshops and conferences.

MOUNIR HAMDI (Fellow, IEEE) is currently the Founding Dean of the College of Science and Engineering, Hamad Bin Khalifa University (HBKU). He is an IEEE Fellow for contributions to design and analysis of high-speed packet switching. As the Founding Dean of the College of Science and Engineering, he led the foundation of 15 graduate programs and one bachelor's program and all the associated research laboratories and activities. Before joining HBKU, he was the Chair Professor at The Hong Kong University of Science and Technology (HKUST), and the Head of the Department of Computer Science and Engineering. From 1999 to 2000, he was a Visiting Professor at Stanford University, USA, and the Swiss Federal Institute of Technology, Lausanne, Switzerland. His research interest includes high-speed wired/wireless networking in which he has published more than 400 research publications, received numerous research grants, and graduated more 50 M.S./Ph.D. students. He is/was on the Editorial Board of various prestigious journals and magazines, including IEEE Transactions on Communications, *IEEE Communication Magazine*, *Computer Networks*, *Wireless Communications and Mobile Computing*, and *Parallel Computing*. In addition to his commitment to research and professional service, he is also frequently involved in higher education quality assurance activities and engineering programs accreditation all over the world.

MONCEF GABBOUJ (Fellow, IEEE) is currently a well-established world expert in the field of image processing, and held the prestigious post of Academy of Finland Professor, from 2011 to 2015. He has been leading the Multimedia Research Group for nearly 25 years and managed successfully a large number of projects in excess of 18M Euro. He has supervised 45 Ph.D. theses and over 50 M.Sc. theses. He is the author of several books and over 700 papers. His research interests include big data analytics, multimedia content-based analysis, indexing and retrieval, artificial intelligence, machine learning, pattern recognition, nonlinear signal and image processing and analysis, voice conversion, and video processing and coding. He is a member of the Academia Europaea and the Finnish Academy of Science and Letters. He is the Past Chairman of the IEEE CAS TC on DSP and a Committee Member of the IEEE Fourier Award for Signal Processing. He served as an associate editor and the guest editor for many IEEE and international journals.