Information Fusion on Delivery: A Survey on the Roles of Mobile Edge Caching Systems

The-Vinh Nguyen[a], Anh-Tien Tran[b], Nhu-Ngoc Dao[a], Hyeonjoon Moon[a], Sungrae Cho[b]

[a]*Department of Computer Science and Engineering, Sejong University, Seoul 05006, South Korea*
[b]*School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, South Korea*

Abstract

Edge caching, which offers application versatility and a range of benefits, has exerted a significant impact on the adoption and development of fifth-generation networks and applications. While many extensive studies on edge caching have been proposed to improve certain system features, a thorough tutorial providing insight into the role of edge caching on mobile data traffic processing is still required. Motivated by these concerns, this study was designed to offer an exhaustive assessment of mobile edge caching. To clarify the role of mobile edge caching techniques, a systematic overview of the state-of-the-art caching models and operations is provided. Subsequently, a complete set of performance indicators is extensively investigated, including hit ratio, storage efficiency, energy efficiency, spectrum efficiency, service availability, and latency, to comprehensively examine each of the caching policy goals. Furthermore, an inquiry into the aforementioned metrics is conducted using popular technological methodologies, such as machine learning, game theory, and optimization techniques. In addition, common use cases and applications for the observation and assessment of caching methods in practice, are described. Finally, the remaining research challenges and future directions of edge caching are discussed.

Keywords: Mobile edge caching, information fusion, digital era, cloud computing

1. Introduction

The last decade has witnessed a dramatic increase in mobile data traffic while fashioning a trend for the foreseeable future. The exchange of data and information is expected to be ultra-high-speed, exuberant, and secure. According to the Cisco Global forecast highlights report [1], internet users will reach 5.3 billion with 29.3 billion connected devices by 2023. Thus, a massive increase in mobile data traffic is anticipated. As for fifth-generation (5G) capability, it is expected that more than 10% of the worldwide mobile devices will have access to the network. Although 5G is capable of offering 1000 times higher throughput, sub-millisecond service latency, and up to 90 percent total energy savings [2], the technology is facing key performance challenges, such as throughput, latency, and energy efficiency, to accommodate the increasing mobile traffic. Thus, to accommodate the growing traffic volume, advanced connectivity, such as beyond 5G and 6G in the future, and the expansion of the system internal capacity are critical. Furthermore, a large amount of popular content is being requested repeatedly and asynchronously, generating in turn a large amount of redundant data, causing a waste of computational resources and energy over networks [3].

Edge caching appears to be a feasible solution for the enhancement of many aspects of networking, such as capacity, energy efficiency, adaptability with diverse applications, and experience/data trade-off. Cache techniques are thus feasible choices for concurrently resolving these issues with data processing and information fusion in network delivery. Conceptually, the caching technique is the process of temporarily storing the copies of frequently used data in an easily accessible location, making the data always available in time with better accessibility. Thus, time and resources are saved because users do not have to request the data again from the original source. Moreover, mobile edge caching systems [4] have been considered in problems related to capacity, data processing speed, and cost reduction. These improvements are critical to the fact that most of the data or content in networks are reinstated asynchronously by many user equipment (UEs) [5].

The ever-growing distributed data render conventional cloud computing inadequate for the transport of data over a congested backbone network to a remote cloud, e.g., macro base station (MBS) [6]. Cloud computing is constrained by unpredictable network latency and high bandwidth, making it vulnerable to the demanding requirements of latency-sensitive applications [7, 8, 9]. Under these scenarios, multiaccess edge computing extends the client-server architecture by introducing intermediate components located at the network edge to improve application responsiveness and proximity to UEs [10]. As a major function of multi-access edge computing platforms, multi-access edge caching (MEC) utilizes the storage capacity of BSs

Email addresses: nguyenthevinh@sju.ac.kr (The-Vinh Nguyen), attran@uclab.re.kr (Anh-Tien Tran), nndao@sejong.ac.kr (Nhu-Ngoc Dao), hmoon@sejong.ac.kr (Hyeonjoon Moon), srcho@cau.ac.kr (Sungrae Cho)

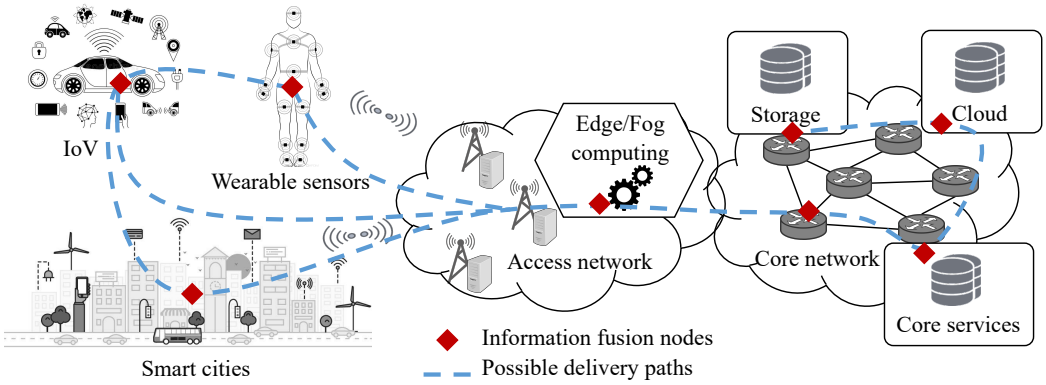

Figure 1: Information fusion on delivery through mobile networks.

throughout the network to perform content placement during off-peak hours, thereby lessening fickle network traffic and reducing congestion and latency. As a practical example, Xunlei, one of the largest content download service providers in China, has adopted a new service that leverages user bandwidth and storage capacity for the implementation of edge caching [11]. Xunlei provides cloud processors (CPs) with edge resources to allow content replication and supply to neighboring UEs. The emergence of MEC technologies has disrupted the traditional notion of mobile radio access networks.

1.1. Motivations

Fig. 1 presents an overview of mobile information fusion on delivery through multiple communication components of mobile networks ranging from access to core tiers. Depending on the specific purpose, services, and requirements, information can be processed either partially or completely at various fusion nodes, i.e., MEC servers. The contents cached on MEC servers are diverse and originated from a variety of sources and nodes, resulting in a massive volume of cachable data. In this context, the information fusion techniques are leveraged to combine diverse sources of information in order to obtain the most dependent, trustworthy, and accurate information. Information fusion on MEC, in particular, combines various types of information such as network state, content popularity, and user preferences, as well as finds and analyzes a huge amount of contents, to selectively cache contents of the highest quality. Furthermore, the introduction of machine learning and artificial intelligence techniques has improved MEC information fusion performance by adaptively learning user information for future operations [12]. MEC information fusion has undoubtedly improved information sharing capabilities and user satisfaction.

MEC transforms mobile access infrastructure into powerful computation and storage entities, providing cloud computing with benefits and capabilities at the edge of mobile networks [13]. MEC can provide pervasive and agile computation, augmenting services for mobile users, irrespective of time and location. It is distinguished by a diminutive latency and high bandwidth, and offers access to real-time radio deployment of applications and services [10]. Furthermore, MEC has improved features, compared to traditional network infrastructures, which enable the performance of specific tasks that could not be achieved before, for example, application-aware performance optimization, big data analytics, and decentralized content caching [14]. The advent of MEC inspired the research community to develop MEC-enabled network models such as information-centric networks (ICNs), heterogeneous networks (HetNets), cloud radio access networks (CRANs), content delivery networks (CDNs), and device-to-device (D2D) networks. Each network model has distinctive properties and features; hence, the caching mechanism should be modified to adapt to the targeted network model [15]. However, these works do not clarify the dynamics of caching use cases and corresponding algorithms at each stage and location of information fusion of delivery through the networks. These key aspects are the prime motivation for this survey.

1.2. Contributions

Overall, the major contributions of our work can be summarized as follows:

- First, a reference framework of state-of-the-art MEC systems is provided to support information fusion on the delivery throughout networks. In particular, caching mechanisms and their properties were investigated to clarify the advantages and benefits MEC provides to the user data. Additionally, several caching strategies for effective content caching were mentioned with the aim to provide readers with a brief overview on the basics of MEC working principal.

- To technically evaluate the roles of MEC in this context, cutting-edge MEC solutions are investigated following a systematic taxonomy of performance metrics such as hit ratio, storage efficiency, energy efficiency, latency, spectral efficiency, and service availability as well as their variants. Each metric was followed by current optimization efforts, illustrating the progress in increasing overall performance of MEC.

- Next, effective caching modeling analyses to achieve the aforementioned performance objectives are reviewed. Here,

the caching modeling techniques are classified into information theoretic, game theoretic, and machine learning formulation categories. The integration of different caching models with the emerging intelligent approaches can significantly improve caching properties.

- Prime application scenarios of MEC for information fusion are provided to support data services and recent updates on practical case studies in the field are described. Throughout this part, the objectives are to highlight the vast potential of MEC, demonstrating the beneficial impacts on human modernization, hence encouraging efforts on perfecting this technology.

- Finally, open challenges are highlighted to drive future research optimizing system performance to flexibly adapt to dynamic changes in environment conditions in different network scenarios as well as increasing user demands.

The remainder of this paper is organized as follows. First, existing works on developing the caching framework and operations are investigated and the main components of a typical caching policy are clarified. Next, state-of-the-art caching proposals are taxonomized by referring to the evaluation metrics of caching performance, including hit ratio, storage efficiency, energy efficiency, latency, spectral efficiency, and service availability. Subsequently, different approaches to caching models are classified from various perspectives. Then, typical application scenarios are analyzed to reveal caching potentials. Finally, open challenges and directions for future research on cache algorithms are discussed to conclude the paper. For convenience, key abbreviations are alphabetically listed in Table 1.

Table 1: Key abbreviations.

Abbreviation	Description
ABR	Adaptive Bitrate
AP	Access Point
BS	Base Station
CCU	Caching Control Unit
CN	Core Network
CRAN	Cloud Radio Access Network
CS	Content Server
CU	Cache Unit
DL	Deep Learning
DNN	Deep Neural Network
DQL	Deep Q-Learning
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
EE	Energy Efficiency
EPC	Evolved Packet Core
GT	Game Theory
HetNets	Heterogeneous Network
ICN	Information Centric Network
ILP	Integer Linear Problem
ISP	Internet Service Provider
LCD	Leave-Copy Down
LCE	Leave-Copy Everywhere
LFU	Least-Frequently Used
LRU	Least-Recently Used
MBS	Macro Base Station
MEC	Mobile Edge Caching
MIP	Mixed-Integer Problem
ML	Machine Learning
MNO	Mobile Network Operator
PPP	Poisson Point Process
QoE	Quality of Experience
QoS	Quality of Service
RB	Resource Block
RL	Reinforcement Learning
RR	Randomized Replacement
SA	Service Availability
SBS	Small Base Station
SE	Spectral Efficiency
SINR	Signal-to-noise-ratio
SPS	Service Provider Servers
TL	Transfer Learning
UE	User Equipment
VoD	Video-on-Demand
VN	Vehicular Network

2. Caching at the Wireless Mobile Edge

In traditional caching mechanisms, reconfiguration is required to exploit caching technology in MEC-enabled networks. Traditional caching schemes do not carefully consider the characteristics of wireless networks, such as dynamic traffic load and interference [9]. The advantages of MEC, as explained in Section 1, fit perfectly with the caching mechanism. Specifically, caching storage, that is, cache memories installed at BSs, should be deployed close to UEs to reduce the transmission distance when a UE requests cached files. These close-range transmissions avoid interference reduction, alleviate spectral congestion, construct reliable communication links, and relax the waiting time of the UE. Second, the location awareness of MEC provides reliable input data for cache policies that predict the UE request pattern and mobility. The caching policies release optimal predictions to indicate which files should be pre-fetched from the CS during off-peak hours, that is, when the number of UE requests is remarkably low. These files are expected to be repeatedly and asynchronously demanded by UEs, which means that duplicated file transfers will not occur. Interestingly, the caching policies would be an optimal technique for MEC-enabled networks because they solve the bottleneck of fronthaul traffic among macro BSs (MBSs), small BSs (SBSs), and UEs as well as backhaul traffic when popular content only needs to be sent once to local BSs rather than multiple times.

There are three typical scenarios for content transmission in MEC-enabled cache-enabled architectures [16].

- If the material requested by a UE has already been cached in the local BS to which the UE belongs, the content is provided directly from the BS to the UE, without incurring additional transmission costs.

- If the requested material is not found in the local BS but has been pre-fetched by another BS in the decentralized caching domain, it is transferred from that BS to the UE via the local BS, incurring low transmission costs.

- If the requested material is not cached in any local BS, the pull request is routed to the CS, incurring a signifi-

cantly high overhead. This is sometimes referred to as the worst-case scenario.

After this section, the readers are expected to have an intimate acquaintance with the main components of caching mechanism, including *(i)* cacheable content, *(ii)* caching locations, *(iii)* caching phases, and *(iv)* caching strategies.

2.1. Cacheable Contents

The main purpose of caching is to allow objects and content to be reused. A series of theoretical studies have been conducted to study cacheable content detection in MEC-enabled networks. In particular, one key area that has been studied is caching policies, which have as mission the detection of cacheable objects to store in resource blocks (RBs) inside storage units in cache units (CUs), because the performance of any caching algorithm is inherently related to the nature of the content. The cacheable contents are interpreted by a certain segment of UEs as high popularity and frequently requested demands, which rely entirely on UE behavior [17]. For instance, in modern social networks, cacheable content is regarded as movies and video clips [18]. Although the UE request patterns are usually indeterminate and non-stationary, modern caching algorithms are still capable of properly approximating the pattern and returning some remarkable reactions when dealing with real mobile data traffic. A probabilistic survey conducted by Tauberg *et al.* [19] indicated that popular media content follows a power-law distribution. This observation supports the argument that the majority of UEs request only a small percentage of available content, intuitively referenced as trends or high-popularity content. Therefore, these contents should be determined and cached as close as possible to the UEs in caching policies.

However, not every object can or should be cached. Due to the limited computing and storage resources of BSs, only a limited amount of content is allowed to be cached at the same time. Objects including information objects such as interactive applications, security information, gaming, voice calls, and remote control signals are not reusable and cannot be cached [20]. Another important consideration is that the cache content may become stale and must be updated so that it is consistent with the content of the server at the origin. Through simulations, Fricker *et al.* [21] demonstrate that some mediums, such as video-on-demand (VoD), are better cached by the MEC, while others such as file sharing should be allocated at the CU of the CS.

2.1.1. Content Popularity

Content popularity is assessed by the number of requests for a certain content divided by the total number of requests from users, generally acquired for a specific location during a defined period [22]. Improving the prediction of content popularity can enhance the efficiency of caching policies [23, 24, 25]. Content popularity changes both spatially and temporally [26]. Using empirical analysis, several video characteristics retrieved from popular video service providers such as YouTube, can be used to determine the overall popularity distribution, the distribution within each video genre, the correlation of the popularity with the age of the video and temporal locality [27]. Interestingly, only 10 percent of the online videos account for nearly 80 percent of the views, while the remainder of the videos account for only 20 percent of the views [28].

The popularity of content has been reported to be amenable to the Zipf distribution [29]. The distributions of files in the web proxies are examples of such a distribution in the real-world [30, 31]. Hence, popularity distribution learning determines the distribution of content popularity as a uniform paradigm, and therefore attempts to optimize the caching decision [32]. As for the working principle, α is the exponent characterizing the Zipf distribution [21], where $\alpha \to \infty$ indicates a heterogeneous distribution, whereas $\alpha \to 0$ makes the distribution more homogeneous. Higher α results in fewer contents, account for most of the requests. Meanwhile, a set of power law distributions [33] can be used as a parameter of the content catalog size and skewness. As reported by Wang *et al.* [34], the Zipf popularity distribution determines the request rate if the size of the requested content is constant. Another point worth mentioning is that if the popularity of the files is the same, the request pattern varies less. Consequently, the worst-case performance may be slightly different from the average case [35].

In practice, content popularity is complex, heterogeneous, and cannot be obtained in advance, given the content dynamics and UE mobility in mobile networks [23]. In addition, content popularity on edge devices has high randomness, leading to inaccuracy in predicting popularity and worsening of the cache performance [32]. In the short term, the dynamic essence of UE-BS association in mobile networks makes the accumulation of sufficient data at the MEC extremely challenging. To clarify this point, the example in [36] is used to compare a D2D wired connection and a wireless mobile network. Despite developments in UE mobility and resources, mobile network cache size is small, leading to insufficient probability in predicting content popularity compared to wired devices.

2.1.2. User Preference

User preference is defined as the likelihood of content requested by a particular user within a specific time period [22]. Indeed, it is natural for UEs to have a strong preference for specific content categories [37] reflecting a certain UE propensity. User preferences may have unintelligible differences depending on their contexts, such as geographic location, personal characteristics (e.g., age, gender, personality, mood), or device characteristics [38]. Most recent studies on proactive caching presume that the popularity profile of content items is perfectly assumed, or denoted based on the Zipf model or its variants [16, 39]. Based on these assumptions, caching algorithms usually model the UE request pattern as a probabilistic parameter and attempt to determine the caching policies optimally. Qin *et al.* [40] developed an evolving social network based on affiliation networks that could scale immediately and rapidly when a new UE or new content enters the network. Guan *et al.* [32] proposed a scheme called *PrefCache* using a preference learning approach to learn user video preferences in real time, improving the hit ratio to 12%, and helping to save 92% memory / 98% CPU overhead.

2.1.3. Prediction Uncertainty

The performance of proactive caching algorithms is affected by the prediction accuracy of the content popularity and user preference. In particular, erroneous information affects the likelihood of locating the requested files in the cache, which is referred to as the *hit ratio*. This metric usually reflects the accuracy of caching performance, as its value is proportional to the number of cached contents requested by the UE in real mobile data traffic. Prediction techniques for the estimation of content popularity profiles usually require a large amount of aggregated data to obtain more accurate results. The prediction uncertainty is also affected by UE mobility. The mobility models represent the movement and changes in the location, velocity, and acceleration of UE over time. Xu *et al.* [41] studied how uncertainty in prediction can affect cache system performance and proposed a generative adversarial network (GAN) scheme as a solution to 5G-enable MEC. The results show that their algorithm outperformed other state-of-the-art algorithms by 15%. In another study [42], uncertainty in popularity prediction significantly affects the performance of edge caching time, especially in video transmission. A Markov-modified caching strategy was proposed, which could be activated when the number of user historical access records was not large. Using the proposed scheme, the hit ratio, accuracy, and speed were improved.

2.2. Caching Locations

The performance of the data retrieval response time in caching can be affected by the distance between data and user. As mentioned, the caching technique minimizes the transmission of data over the MEC and allows data to be served immediately upon request, thereby directly improving response time and bandwidth utilization. By optimizing the location for data caching, service delay can be reduced, making the location optimization problem an important aspect in the implementation of 5G and 6G technologies [43, 44].

Given the geographical and temporal variability of mobile data traffic, the global content distribution may not always be available to satisfy all local demands. Therefore, caching locations should predict local content popularity for a *proactive* cache content placement in lieu of a global one. Insights into the deployment of caching locations can be acquired by defining performance metrics, such as service availability, average delivery rate, skewness of content popularity, storage size, and target signal-to-interference-plus-noise ratio (SINR) [45]. The caching locations over the edge networks are discussed as follows:

- **MBS Caching:** In heterogeneous networks, MBSs provide the largest coverage areas and ample storage resources; thus, they can serve numerous users. Therefore, caching at the MBS provides a better cache hit ratio, making MBSs great candidates for implementing an edge cache. For example, Ahlehagh *et al.* [37] demonstrated the effect of MBS caching in video streaming services by showing that significantly boosting the video quality can alleviate the stalling probability of videos. Leveraging the wide connection range of MBSs, vehicular networks are another potential field for future development. In [46], a proactive caching approach using federated learning was proposed to resolve the problem of vehicle location sensitivity during the caching process. MBSs act as routers between the internet and roadside units (RSUs) to manage RSUs cache resources.

- **SBS Caching:** SBSs are considered a promising infrastructure to accommodate the exponential growth of wireless traffic in future 5G networks. They are densely deployed within macro-cells according to a PPP and serve the UEs from the local caches or CN through a finite rate backhaul, helping achieve high-density spatial reuse. Furthermore, they are characterized by a large storage capacity and are deployed relatively close to end-users. They have been identified as great caching candidates in wireless networks because of their ability to reduce backhaul traffic and minimize content access latency. Caching at SBSs often results in higher energy efficiency (EE) because SBSs have more opportunities to idle, having low transmission and circuit powers, although the cache capacity of each SBS is smaller than that of each MBS. In the work of Bastug *et al.* [45], caching at SBSs assures certain levels of service availability by simply increasing the number of SBSs or the total storage size. In [47], a software-defined networking (SDN) incentive caching framework was proposed for a 5G vehicular network, formulated as the Stackelberg game. In this scheme, SBS allows vehicles to communicate with each other via V2V by offloading cellular core links, enabling the caching strategy to earn more SBS rewards, leading to higher caching utilization. An application of SBS in a 6G environment was introduced for autonomous driving in [48]. The authors pointed out the weakness of SBS in 6G development and vehicular networks, which is the mobility of users requiring a faster shift among SBSs. This can cause a longer delay in content delivery, making it insufficient for QoE maintenance. One feasible solution is to have a cache strategy with a reinforcement learning approach to select the cached content in the local cache, edge server, and SBS.

- **User Devices:** Modern UEs are continually equipped with more computational resources and larger storage capabilities (e.g., several gigabytes). This presents opportunities for storing popular content that can be reused by nearby edge users. The QoE of users can be significantly improved by caching contents in UEs when device-to-device (D2D) transmission is preferred over UE-BS transmission. The D2D is promising in relaxing traffic congestion and shortening the transmission distance to increase the spectrum efficiency (SE) and decrease the latency of UEs. For example, a caching D2D-based communication scheme was proposed in [49], which considers the social relations between users with common interests, thus defining the cacheable contents to be placed at UEs in

off-peak hours. This example illustrates the benefits of caching in UEs.

2.3. Caching Phases

There are three main phases in the caching mechanism: cache placement, cache delivery, and cache replacement [18]. In this subsection, the differences between these phases are analyzed.

First, the network is non-congested throughout the cache placement phase, and the system is primarily restricted by the size of the cache memory [27]. BSs cooperatively cache content into their own respective storage. In the next phase of cache delivery, the network is crowded and the system is primarily confined by the content request rate from UEs. The local BSs search for corresponding content and deliver them to the UE. If no such content is available, the BSs fetch them from the core network and transmit them to the UE. Finally, the replacement phase immediately occurs when a BS detects the content being transmitted through it. The caching policies determine the new content to cache and select the cached content to be replaced with the new content. The caching phase performance is evaluated by the status of system parameters, for example, fronthaul and backhaul links, energy, and storage.

A majority of recent studies on edge caching have mainly focused on solving the problems of caching phases. In particular, the caching algorithms proposed in these papers targeted some important features of the caching phases to optimize the targeting metrics. Theoretically, caching decisions are often characterized as binary variables (0 or 1) to train the cache units (CUs) to either cache or skip the contents, respectively. An effective caching decision can significantly reduce energy waste, transmission latency, and cost.

2.3.1. Cache Placement Phase

The objective of cache placement is to maximize the cache hit ratio [50]. In other words, content is placed in the cache with the assumption that the content has a high probability of being requested by users and it will be available when a pull request is made to the cache. The cache placement consists of subproblems, such as deciding the size and position of each cache, choosing the material from a library to store at selected nodes, and determining how to download content to these cache nodes [24, 25, 51, 52]. The ultimate goal of caching control units (CCU) is to minimize the cost of cloud services by intelligently selecting the cache contents. With predictable content popularity, through data analytics and CCUs, popular content can be cached locally before requests from UEs arrive at the caching locations [22]. Cache-enabled networks begin the cache placement phase only when there is a small number of requests, that is, off-peak hours. During the cache placement phase, the spectral resources, computing devices, and storage are frequently assumed to be free and without interference. The expensive computational elements of caching algorithms, such as the prediction of future trends of content requests, are also suggested to be implemented in this phase. Learning-based algorithms, such as machine learning or deep learning, can produce excellent performance but require a large amount of data and time for training.

As shown in [53], cache placement and bandwidth (BW) are collaboratively optimized in a frequency-division multiple access (FDMA) setup to minimize edge server energy, computational cost, capacity, and latency. The authors used Lagrangian duality and the ellipsoid method to solve the resource allocation problem, and then used a heuristic algorithm to update the cache placement. In [54], the utility is resolved in the cache placement phase of a large-scale information-centric network (ICN). A distributed cache placement scheme was devised with the goal of pushing popular material to the edge network while maintaining less popular content at the core. A collaborative technique was also suggested for retrieving material from the content-store of the nearest neighbor as well as a cache replacement policy depending on content popularity.

2.3.2. Cache Delivery Phase

With cache delivery, or content delivery, the main issue is determining how to deliver content to a user who requests it. During the content delivery phase t, each user requests a selection of files from the file library $\mathcal{F} \equiv \{1, 2, ..., F\}$ locally. The edge nodes determine the availability of the requested file. If the requested file has already been cached locally, it can easily be delivered to users almost without cost. On the other hand, the SBS must retrieve the requested file from the cloud by using a backhaul link if the file is not already cached, thus resulting in a significant cost due to potential electricity price surges, processing costs, or a large delay, leading to a drop in QoS and user dissatisfaction [55, 56, 57]. A common assumption is that the files in the system can be divided into equally sized chunks, and the transmission of one file can be transmitted in a single slot. In MEC-enabled networks, D2D caching and multi-hop caching are the best networks for enhancing the cache delivery phase because they enable neighboring UEs to create peer-to-peer connections, and disseminate cached material as needed by other UEs.

2.3.3. Cache Replacement Phase

To ensure that the cached content is always accessible and satisfies user demands, the CCU replaces existing content in the cache by caching new ones prior to UE requests and removing the no-longer-popular contents from the cache as needed [58, 57, 59]. Each replacement algorithm can be considered as a trade-off between cache hit ratio and delay. The caching policies must justify the performance of the cache-replacement phase when new files are requested while prioritizing the most popular content. Caching policies can easily adapt to rapid changes in the content distribution owing to a well-structured cache-replacement phase. Conventional replacement algorithms such as least frequently used (LFU), least-recently used (LRU), leave-copy everywhere (LCE), leave a copy down (LCD), first-in-first-out (FIFO), and randomized replacement (RR) are often treated as baseline algorithms to examine the efficacy of newly proposed algorithms because of their simplicity [60]. The details of these baseline algorithms are discussed in the next subsection. As mentioned in Section 2.2, [46] also incorporates a mobility-aware cache replacement policy, which enables net-

work edges to add/evict content based on vehicle mobility patterns and preferences.

2.4. Caching Strategies

The cache content can directly influence the performance of the content delivery network, thereby affecting the QoE of users. Learning what, when, and how to cache, or having an effective caching policy, is a critical issue in mobile edge caching [50]. It is crucial to estimate the gain behind a content by assessing its present popularity, projected popularity, storage size, and placement of existing clones across the network topology. In this subsection, the categorical classes of caching algorithms are reviewed before delving into their definitive metrics.

2.4.1. Conventional Caching Strategies

Some baseline caching policies can be listed as LRU, LFU, LCD, LCE, and RR. Many of them have long been implemented in a real system because of their simplicity and reasonable performance. The common values of these baselines are easily implementable and have low complexity. However, all of them have critical problems that cannot be addressed, and hence they need to be replaced or modified in future networks.

- ***LRU:*** LRU believes that the probability of recently requested data to be repeatedly requested in the future is higher than other data. Therefore, when the cache storage reaches its maximum capacity, it replaces the least recently referenced objects [61, 62, 63]. However, there is not enough information for the LRU to decide which content to drop, limiting it only to the time of the last reference.

- ***LFU:*** LFU uses the object popularity as the primary factor [64]. When the cache storage is full, it discards the content that has the lowest object popularity. However, new contents that just entered the cache are likely to be eliminated because of their low early counter, even though they might be utilized often subsequently.

- ***LCD:*** In LCD, the requested content is only saved at the cache location from which the requests came, i.e., local SBSs.

- ***LCE:*** In LCE, objects move gradually from the CS towards the UEs and are stored at any cache location they propagate through. The CUs in this caching algorithm can quickly incur an overhead.

- ***RR:*** RR randomly discards any content in the cache storage when space is running out, and thus obviates the need for any information on that content. Intuitively, because RR has no concrete indicator, the performance is usually poor due to instability [65].

2.4.2. Centralized Caching vs. Decentralized Caching

Recent studies do not provide enough information for a comprehensive overview of centralized and decentralized caching. The choice of caching strategy is made from the perspective of the user (e.g., user data management) or control (e.g., making caching strategy). Centralized caching, as the name implies, requires a central base station to implement caching policies. In addition, the caching replacement and delivery phases in centralized caching schemes are consistent and synchronized [66]. For example, when a new UE is connected to a network, all cache must be reconfigured. An advantage of this model is its simple implementation and homogeneous connection. However, the model is vulnerable to security threats because all caches are under a central BS, making them less energy efficient and dependent, leading to weaker performance. By contrast, decentralized caching, that is, distributed, operative caching, typically saves cache space by avoiding replication of cache contents in neighboring caches [67, 68].

Compared to centralized schemes, decentralized ones have a random placement phase; cache content is independent of other caches, and therefore has higher flexibility and consumes less energy during configuration. Recently, federated learning (FL) models have been widely exploited, in which centralized control and decentralized agents are combined to accelerate learning convergence and prediction accuracy. In [69], Ji *et al.* proposed a decentralized random caching scheme, using maximum distance separable (MDS) coding, to ensure that all files can be recovered by the (coded) symbols cached into the network, with high probability. When the network size increases, this scheme achieves order optimality. Overall, the decentralized random caching scheme appears to be practical because all UEs cache randomly and independently their assigned fraction of coded symbols of the library files, without considering whether the symbols have already been cached by other UEs. In a recent study on both centralized and decentralized caching [39], a deep reinforcement learning framework with Wolpertinger architecture was introduced to maximize the cache hit ratio in centralized schemes, and the cache hit rate as well as transmission delay in decentralized schemes.

2.4.3. Coded Caching

Data types are crucial to any cache policy and are separately divided into coded and uncoded data. The caching policy problem is framed as an integer optimization problem for uncoded data, which is frequently an NP-complete problem. On the other hand, the caching policy for coded data is considered an optimization issue in linear programming that can be addressed in polynomial time. Uncoded transmission consists of packets from the same file for the use of each channel, whereas coded transmission is a combination of multiple packets from different files [70]. Without loss of generality, it is assumed that the algorithms work with coded data as coded caching algorithms and otherwise as uncoded caching algorithms.

Coded caching is widely used in other fields of research based on caching techniques as a class of communication methodologies [70, 71]. The idea of caching was introduced in the fundamental work of Maddah *et al.* [72] as a new term for information theory community, *coded caching*, which promises unprecedented gains. In the proposed paradigm in Fig. 2, an unbounded K number of users can each accommodate up to M cache files, connected to a single server via a shared-link

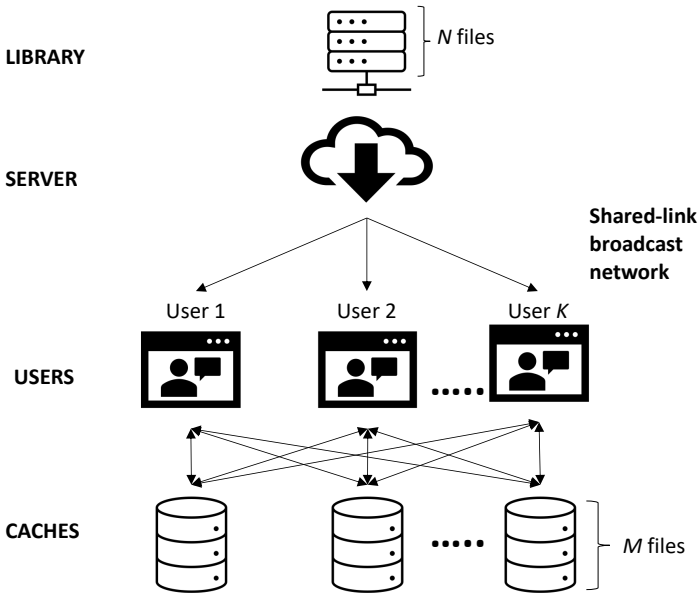

Figure 2: Coded caching scheme [73].

broadcast network. This server is accessible to a library which contains N equal-length segments of cacheable content. Coded caching works in the first two phases of the caching, placement, and delivery phases. First, during the replacement phase, users cache the received files sent by a server. These files are random and do not know the user preference or demand. Second, in the delivery phase, each user can demand a file. Based on these demands and caches, the server broadcasts coded messages to all users via a shared link so that it can satisfy all user demands. The goal of coded caching is to reduce the worst-case number of transmissions in the delivery phase, normalized by file size (referred to as worst-case load, or load), among all potential demands [73]. The load achieved by coded caching (R), proposed by Maddah [72], is formulated as follows:

$$R = \frac{K(1-\frac{M}{N})}{\frac{KM}{N}+1}, \quad \forall M = \frac{Nt}{K} : t \in \{0, 1, \ldots, K\}. \tag{1}$$

Furthermore, coded caching uses fewer RBs to serve multiple concurrent users with different video demands compared to uncoded caching. Niesen *et al.* [74] presented practical and numerical examples to illustrate the effectiveness of coded caching via multi-casting and applied it to delay-sensitive content. To evaluate the effectiveness, in terms of the spectral efficiency of coded caching algorithms, many authors [74] attempted to maximize the global coding gain, which is the ratio of the bandwidth between the uncoded and coded schemes. Some recent works on coded caching, such as [75], improved the caching strategy in ultra-dense networks with maximum-distance separable (MDS) coding by using reinforcement learning approaches, and Wan *et al.* [76] enhanced the privacy of users by introducing a novel shared-link caching model with private demands.

2.4.4. Reactive Caching vs. Proactive Caching

The current ***reactive caching*** networking model, in which UE traffic requests and flows must be served immediately upon being dropped or upon arrival, result in outages. Consequently, the current small-cell networking model is incapable of addressing peak traffic needs. Thus, large-scale implementation is dependent on costly site acquisition, installation, and backhaul expenditures. These flaws are expected to worsen as the number of connected devices grows and ultra-dense networks emerge, putting the present cellular network infrastructures under strain [22].

According to replacement algorithms, a reactive caching policy chooses whether to cache the item after it has been requested. Popular reactive caching strategies, the least recently used (LRU) and its modified version, probabilistic LRU (p-LRU) [61], are made available to tackle time-varying content popularity in a heuristic approach. The LRU replaces the cache element with a new file, which is not used or requested for a long time, as determined by the insertion time parameter. The most recognizable advantage of this algorithm is the ease of implementation. In [77], as an optimum multi-level cache content placement approach, a non-cooperative Hierarchical Reactive Caching (nCHRC) algorithm is suggested to study the effect of popularity time on cache hit ratio, backhaul traffic, and delay.

Proactive Caching is a significant enabler of 5G wireless networks through the deployment of small cell networks. The proactive strategy takes advantage of the existing MEC and entails the development of predictive radio resource management techniques for 5G network optimization. In mobile networks, proactive caching relies on mobility prediction to locate the UE next location, and hence, the SBS must pre-fetch the content. During peak traffic periods, poor backhaul link connections can quickly become overloaded, reducing the QoS of UEs [78]. One way to overcome this constraint is to transfer the excessive load from peak to off-peak hours. Caching achieves this transition by retrieving the expected popular material, such as reusable video streams, to store them in SBSs supplied with memory units and reuse them during peak traffic hours [79].

In comparison, Bastug *et al.* [22] examined two cases that used the spatial and social structure to demonstrate the effectiveness of proactive caching. For backhaul congestion, they suggested using a technique in which content is cached *proactively* during off-peak demand based on content popularity, user correlations, and content trends, taking advantage of the social network structure by predicting the set of significant users who would cache strategic material. The spectral efficiency and hit ratio rate are increased by 22% and 26%, respectively because of the simulation setup, compared to *reactive* algorithms. In [37], the performance of reactive and proactive caching at the MBS was investigated. A video-aware backhaul and wireless channel scheduling approach, along with edge caching, was presented. The results show that the video capacity may be considerably improved while the video stalling likelihood is minimized. However, with prediction error, the hit ratio of proactive caching may perform worse than that of reactive caching [80]. As an interesting suggestion, in [81], Jiang *et al.* jointly lever-

aged both proactive and reactive caching in their proposed policy to maximize cache hit ratio in fog radio access networks (F-RANs). The suggested policy can quickly track multiple popularity trends with spatial-temporal popularity and user dynamics while maintaining low computing complexity.

2.4.5. Cooperative Caching

Cache capacity has always been a critical issue to address. As the number of users increases, and so does the variety of requested content, ***cooperative caching*** appears to be an effective caching solution for reducing content request time and improving user QoE. Recent studies have demonstrated that cooperative caching is the most successful caching strategy for the optimization problem in the cache placement phase and has attracted a lot of attention. The working principle of cooperative caching is to allow edge servers to collaborate with distributing data items via internal connections. Thus, the load capacity of each server is reduced as well as UE access to more content. The goal of cooperative caching is to fully use idle servers while avoiding storing too many redundant data copies with assured data retrieval time.

In [82],the hierarchical cooperative caching problem in fog radio access networks (F-RANs) was investigated to identify the optimal policy. A brainstorm optimization (BSO) using a penalty-based fitness function for individual assessment was proposed to overcome the storage capacity limitation. The request delay for the content was reduced using the proposed scheme. Another work [83], presented a cooperative edge caching strategy based on deep reinforcement learning (DRL) for efficient collaboration among distributed edge servers. The cache hit rate can be enhanced by developing a suitable incentive function and a multi-agent actor-critic method, resulting in greater cooperative caching performance. Cooperative caching can also be applied in vehicular networks because it enables a vehicle to retrieve data from several cache servers simultaneously. In [84], a cooperative caching scheme was proposed for two request types: location-based and popularity-based. Content placement and portions are metrics for delay and cost optimization problems, respectively. This problem is formulated as a multi-objective, multi-dimensional, multi-choice knapsack problem and resolved using an ant colony optimization-based algorithm.

2.5. Summary and Discussion

In this section, four main components of edge caching were thoroughly analyzed, including cacheable content, caching location, caching phases, and caching strategies. In the first part methods for the effective selection of suitable content for caching were presented, based on content popularity, user preference, and prediction uncertainty. A high cache hit ratio is achieved when there is a high probability of caching relevant content, thereby improving the wait time for users and saving energy and computational resources for the caching system. Moving the cache near end-users has also been considered a promising research direction. As the number of UEs with diverse applications rapidly increases, having a cache close to users can leverage UE resources, lowering the computing strain of the base stations, and decreasing latency. MBS has the highest coverage and hence the highest computing power and cache capacity. However, there is a considerable distance between MBS and users, resulting in lengthier wait times for content requests and responses. SBS, on the other hand, has a narrower signal range. The high density and proximity to consumers, make SBS particularly effective for transmission of data. As user devices are becoming more powerful in terms of computation and storage, using UE is an option worth considering. There are three major steps in caching: placement, delivery, and replacement. The goal of the caching phase is to regulate the flow of content in both the cache system and the user perspective. Content is cached to achieve the highest cache ratio in delivering to the user and in refreshing the cache content. Finally, selecting an optimum caching technique may optimize the caching process, significantly enhancing the speed and cache hit ratio; LRU, LFU, LCD, LCE, and RR are examples of typical conventional caching techniques. Some current advanced caching strategies that have been thoroughly researched include decentralized/centralized, coded, reactive/proactive, and cooperative caching. A suitable strategy may be chosen based on the present scenario, purpose, and caching framework. A promising approach for the future is multiple caching algorithms operating together to enhance data sharing and communication across edge servers.

3. Performance Metrics

In this survey, a set of evaluated metrics that are used to characterize the caching algorithms are proposed and a systematic overview of the caching algorithm targets that have been achieved thus far is provided. The set of metrics include *(i)* hit ratio, *(ii)* storage efficiency, *(iii)* energy efficiency, *(iv)* latency, *(v)* spectral efficiency, and *(vi)* service availability, as shown in Fig. 3. The details of each metric and the achievements of the scholars who proposed the metrics are clearly explained in the following subsections.

3.1. Hit Ratio

A cache hit event occurs when a file *j* requested by the user *i* is presented in the cache and can be served immediately. Similarly, a cache hit ratio is the possibility that files requested by users are already cached in the caching space. This criterion illustrates the percentage of cached files used, which is an effective parameter for caching performance evaluation. Table 2 summarizes some recent approaches in the development of caching algorithms using the hit ratio metric.

Roberts *et al.* [91] state that the memory-bandwidth trade-off relies heavily on the hit ratio, which calculates the proportion of downloaded throughput saved by a given-size cache, and evaluates the hit rate using the Che approximation assuming LRU replacement under IRM. In [31], Song *et al.* maximize the hit ratio using a greedy method to identify the best cache location, which is complex and independent of file library size. Optimal cache placement balances the trade-off between channel diversity gain and cache diversity gain. As a result of recent

Table 2: Prime examples of hit ratio studies.

Ref.	Technical approach	Management model	Cache phase	Merits
[17]	Active Learning	Centralized	Cache Placement & Replacement	Improve content popularity prediction speed while maintaining high cache hit ratio.
[22]	D2D caching	Centralized	Cache Placement	Propose proactive networking paradigms to substantially reduce peak data traffic demands via caching strategic contents at both BSs and UEs.
[25]	Deep learning	Centralized	Cache Policy	Apply LV and NN to evaluate, quantify, and predict UE content requests, which are then used to select the most interesting UE content to cache at SBSs. The networks eventually benefit from higher hit ratio, spectral efficiency, and storage efficiency .
[57]	Knapsack	Decentralized	Cache Placement & Delivery	Using Knapsack in conjunction with Zipf distribution to store highly popular materials and to entertain partial queries with significantly improved hit ratio.
[58]	MDP	Decentralized	Cache Replacement	Propose a versatile approximation of ICN caching systems to model, analyze different caching schemes, and acquire useful insights into ICNs.
[68]	Transparent learning	Centralized	Cache Placement	Propose a machine learning framework based on transparent computing, which trains data at the MBS and locates the test models on the client side.
[85]	Vehicular Named Data Network	Centralized	Cache Replacement	Separate messages into three categories, analyzing their respective spatial-temporal characteristics, and implement suitable caching strategy based on analyzed information.
[86]	Gray model	Centralized	Cache Policy	Apply gray model, which has high accuracy in predicting the sequence of content popularity and tracking the trend of UE content interest.
[87]	Deep reinforcement learning	Centralized	Cache Replacement	Require no knowledge of the content popularity distribution. Nevertheless, it improves and stabilizes the long-term cache hit ratio with reduced runtime.
[88]	Iterative	Centralized	Cache Replacement	Improve the LFU algorithm by adding object size to the formula of weight.
[89]	Machine learning	Centralized	Cache Replacement	Utilize the advantages of RNN to predict the next location of UEs as well as the associated content of interest.
[90]	Graph theory	Centralized	Cache Placement	Propose a simple greedy caching algorithm that topologically sorts the UEs, CUs, and BSs to find the shortest content delivery path.

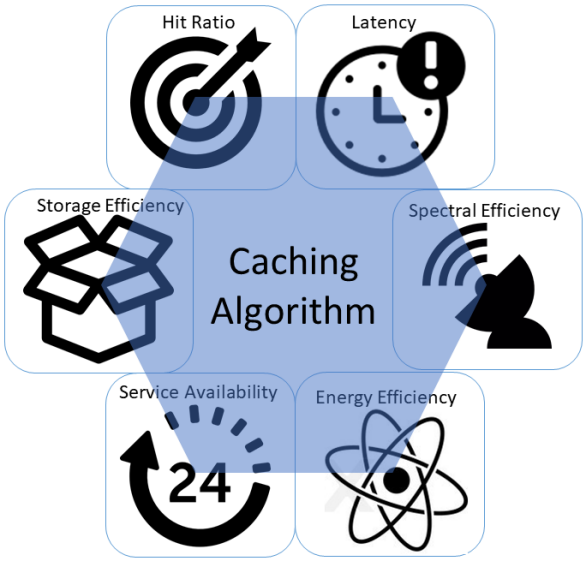

Figure 3: Evaluation metrics for caching algorithms.

advancements in learning techniques, the popularity of content can be anticipated by tracking user request frequency and previous data evaluation [22], and then proactively cached. Based on the observation that UEs are likely to visit certain sites on a regular basis [92], Tang *et al.* [89] use the recurrent neural network (RNN) model to predict the future position and interest of the UE,, based on historical traces, and then update the cached content of SBSs at the projected location to subsequently serve to the UE. The author proposes an equipment-assisted caching multimedia framework to lease and adjust edge IoT equipment adaptively to match the temporal and geographic changes in UE needs. The framework can also be combined with the LRU and LFU algorithms to optimize caching replacement strategy, which considerably improves the hit ratio and reduces data acquisition time.

In [86], Xiaoqiang *et al.* presented a cache policy based on the betweenness centrality and content popularity prediction, by using a gray model [93] to improve the hit ratio, shorten the access distance, and reduce the transmission delay. The betweenness centrality reflects the role and influence of nodes in the network; the higher the betweenness, the more the number of content distribution paths through it, and the more likely it is to get a cache hit. Bernardini *et al.* [94] designed the most popular content (MPC) strategy, a content popularity caching strategy for CCN, where only popular content is cached by nodes. The MPC caches less content while maintaining a higher hit ratio than the default caching strategy of CCN [18]. The hit

ratio grows continuously and consistently over time because it needs to assess popular contents to cache them, as well as to dramatically reduce the number of replicas of elements, which enhances network storage efficiency. In [68], Guo *et al.* proposed a transparent learning-based framework together with an incremental learning process to improve the cache hit ratio. Another recent interesting caching approach proposed by Chen *et al.* [85], which used to increased the average cache hit rate, hop count and cache replacement times in vehicular networking environment. Realizing the insufficient of TCP/IP protocol for data transmission, the authors have leveraged the emerging Named Data Network (NDN) technology to effectively cache contents by explicitly divide forward messages into three types and analyze the spatio-temporal properties of each type. Compared with the existing LCE, Pro(0.7), and Pro(0.2), the proposed solutions achieved 50% better performance in the three mentioned metrics.

In general, the assessment of cacheable material is performed in two parts: whether to cache the arriving content and whether to delete the content that was cached before. These two steps are carried out using the selection algorithm and prediction cache, respectively. In [57], Furqan *et al.* presented a collaborative caching approach that matches each UE request to any of the caches inside the hotspot to reduce the traffic load to the EPC as well as the transmission delay rate and cache hit ratio. The authors also employed the knapsack problem in conjunction with the Zipf distribution to increase flexibility and reduce coding necessity to indicate the rank of extremely popular stored content, which is determined by the number of content requests made by UEs in a time unit. These contents are then selectively cached in the hotspot. In [88], Ma *et al.* proposed a low-latency, high-hit rate cache replacement policy for a web cache which considers time, frequency, and the cost value of Web objects, to design a new weight calculation method based on cost. The algorithm replaces the largest weighting and cost objects when the cache space is about to run out. Bommaraveni *et al.* [17] used an active learning strategy to discover the best content popularity. By forecasting content popularity with high accuracy, new content can be cached in the placement phase, and the system can also learn about user preferences to replace cached contents with new ones in the replacement phase. The results suggest that the popularity of learning new material is increasing at a faster rate while retaining a high cache hit ratio.

3.2. Storage Efficiency

Storage efficiency measures the efficiency of physical storage devices, including UE cache storage [78, 51, 97]. The cache storage size is finite; therefore, caching decisions must be made considering space constraints. This means that as content is placed in the cache; the storage eventually becomes full, leading to the eviction of some objects already in the cache according to certain criteria, such as frequency (LFU-LRU [100]) of access. In practice, there is also a trade-off between delivery delay and storage efficiency. A higher storage capacity indicates a greater burden on the data transmission speed. Therefore, coded caching can be used as a possible solution to increase storage capacity by encoding files into smaller chunks. In [101], the authors tried to find an optimized solution to balance the trade-off: maximize storage capacity while ensuring minimum delay in delivery.

In [78], Sadeghi *et al.* proposed an RL-based caching method that uses a Q-learning algorithm to apply optimum policy live, allowing CCUs at SBSs to learn, track, and perhaps react to the underlying dynamics. Sadeghi *et al.* also provided a linear function approximation to make this approach scalable, reducing the complexity and memory requirements. Assuming UE interest is known, Hachem *et al.* [95] proposed a coded multilevel popularity-aware memory allocation framework that aggregates non-uniform content popularity into no more than three or four different single-level subsystems in order to efficiently address the trade-off between the transmission cost at MBS, and the storage cost at the SBSs and the access cost at the UEs resulting from connections to multiple SBSs. The aforementioned work of Abani *et al.* [51] also resolves the storage efficiency problem by leveraging the ICN flexibility of caching anywhere. The storage redundancy is eliminated, and the cache hit ratio increases. Di *et al.* [96] combined proactive caching rules with offloading decisions and analyzed the amount of *offloadable* tasks to dramatically increase the hit ratio and storage economy, particularly in the case of SBSs with small cache sizes. The cache policy considers the task popularity as well as the sizes of the input and processed output to choose cache tasks with small outcomes.

A cooperative caching strategy was presented in [98] to overcome the problem of low storage capacity in mobile edge caching. Because linking edge nodes in data communication is a mixed integer nonlinear programming issue, a *ICE* Gibbs sampling approach was utilized. Response times are considerably lowered when the cache content is chosen with storage in mind. Furthermore, synchronization of both storage and computing resources improves the system performance. In [99], it was stated that the problem of distributed storage in erasure-coding can result in long latency. A time-to-live (TTLCache) caching framework was proposed by dividing the aforementioned problem into three sub-optimization problems: access optimization, window size optimization, and auxiliary variable optimization. The mean-tail probability was characterized using the TTLCache policy and probabilistic selection. Table 3 summarizes recent studies concerning the storage efficiency.

3.3. Energy Efficiency

The downlink network energy efficiency (EE) was calculated by dividing the average number of bits sent by the average energy used [110, 111], which is comparable to the ratio of the network average bandwidth to the BS average total power consumption. It can also be interpreted as power consumption when battery lifetime of user devices and electricity bills at operation sites are taken into consideration. Optimizing EE in edge caching has been an interesting topic, yet difficult to solve, especially in deploying 5G and 6G scenarios. As caching reduces delay and improves QoE for users, it is computationally expensive and energy intensive. Furthermore, in the deployment of 6G technology, unmanned aerial vehicle base stations (UAV-BSs) have emerged as a feasible way to enable

Table 3: Prime examples of storage efficiency studies.

Ref.	Technical approach	Network model	Cache phase	Merits
[51]	MDP	ICN	Cache placement	Use entropy to measure the uncertainty of MDP-based mobility predictions.
[78]	MDP	HetNets	Cache delivery	Propose a scalable algorithm that enables SBSs to learn, track, and possibly adapt to the underlying dynamics.
[95]	Optimization	HetNets	Cache placement	Propose a scheme that carefully allocates UEs, based on prior knowledge of their interest, and caches SBSs into different groups so that there are no overlaps within a group, and then serves them separately.
[96]	Joint optimization	MEC	Cache placement	Combine caching policy with offloading decisions.
[97]	Approximation	ICN	Cache replacement	Build an eviction policy that mixes LFU and LRU to adapt to ICNs.
[98]	Cooperative service	Mobile	Cache Placement	Reducing workload and increase cache storage for edge nodes by cooperatively sharing content. A two-layer, Gibbs-sampling-based ICE algorithm, was proposed to solve the scheduling problem.
[99]	Joint optimization	Erasure-coded storage	Cache Delivery	Jointly optimize mean-tail latency to improve systems with low storage performance.

Table 4: Prime examples of energy efficiency studies.

Ref.	Application scenario	Network model	Technical approach	Management model	Cache phase	Merits
[55]	Social network	Mobile networks	Reinforcement learning	Centralized	Cache delivery	Propose a RL method that achieves near-optimal performance while keeping the computational costs at an affordable level.
[102]	Broadcasting	HetNets	MDP	Centralized	Cache delivery	Find out the optimal multi-cast scheduling policy, which is adaptive to the request queue state.
[103]	IoT	HetNets	Randomized rounding techniques	Centralized	Cache delivery	Develops an algorithm that works well for the thousands of attendees case, in which the UE requests for content are delay-tolerant [104].
[105]	SBS caching	Hierarchical	Optimization	Decentralized	Cache policy	When the interference level is low, the backhaul capacity is tight, and the content popularity distribution is skewed, the EE gain is substantial.
[106]	Green networks	MEC	Game theory	Centralized	Cache delivery	Build an algorithm that adopts the interference among UEs and has a stable and optimal performance in terms of network overhead.
[107]	Big data	HetNets	Deep reinforcement learning	Centralized	Cache policy	Integrate networking, caching and computing to eventually minimize the total energy waste of green HetNets.
[108]	ICV	Vehicular Network	Optimal stopping theory	Decentralized	Cache delivery	Choosing edge node based on UE geographic location to optimize the distance between cache and UE, hence saving energy consumption.
[109]	Video Streaming	MEC	Stochastic mixed-integer programming (SMIP)	Decentralized	Cache delivery	jointly decide the energy-efficient caching of bitrate-aware files and the scheduling of video requests.

internet access to remote areas [112]. With increasing mobility and computation ability, in addition to the variety of data and content demands of users, it can be predicted that the energy optimization problem will become inevitably more challenging. The most common approach for energy solutions is to optimize the trade-off between energy and other metrics, such as resources, bandwidth, and latency, by using suitable caching strategies.

In [102], Zhou *et al.* formulated the stochastic minimization problem of the average delay, power, and fetching costs (depending on the content size) as an infinite horizon average cost Markov decision process (MDP). In the uniform case, the derived optimal policy has a switch structure such that a content is multicast to all users using the same transmission power and partial switch structure of the nonuniform case. To provide material to various consumers, different transmission powers are necessary. A low-complexity suboptimal cache policy with a switch structure is proposed to reduce computational costs. In [105], Liu *et al.* derived the closed-form expression of the approximated EE, provided the circumstances under which the EE could benefit from caching, determined the appropriate cache size that optimizes the EE, examined the maximum EE gain brought about by caching, and concluded that caching at SBSs provides a bigger EE gain than caching at MBSs. A key observation is that when the interference content is reduced, the backhaul capacity is restricted; the distribution of content popularity is biased; and the EE benefit is substantial. Somuyiwa *et al.* [55] modeled the problem of proactive content caching of a limited CU capacity UE as an MDP problem to maximize the long-term gain of EE with a time-varying channel state. The MDP problem was optimally addressed by a policy representation characterized by low-complexity, cooperating with

the policy gradient RL. In [103], Poularakis *et al.* proposed a model that explicitly addresses the heterogeneity of the SBSs with factors governing cache size and transmission cost (e.g., different energy consumption profiles [113]) and high variation in the request patterns of the UE. Subsequently, Poularakis *et al.* created an approximation caching method with guaranteed performance and a heuristic caching strategy to reduce energy expenses in cases of high demand for delay-tolerant materials. In [107], He *et al.* proposed a framework for integrated networking, caching, and computing to improve the EE of green HetNets. To deal with the high complexity of the system, He *et al.* used a DQN to approximate the Q-value-action function. Energy consumption in edge caching is extremely important when edge users have high mobility, such as in vehicular networks, because they need to track UE locations, optimize the shortest route for content transmission, and ensure network stability. In [108], an edge caching, energy-aware cache node selection framework (OEECS) was introduced to address the energy shortage problem of intelligent connected vehicles (ICVs). As an optimal stopping problem, the technique selects the ideal caching node based on the position of the vehicles. Consequently, less energy is needed for content caching. From a video streaming perspective, Li *et al.* [109] investigated the trade-off problem between energy efficiency and video quality in adaptive bitrate streaming services. Li presented a comprehensive approach to jointly decide the caching of bitrate-aware files and the energy-efficient scheduling of video queries in an MEC-enabled adaptive streaming system by integrating caching, transcoding, and backhaul retrieval. Energy optimization was formulated as a two-stage stochastic mixed-integer programming (SMIP) problem. The simulation results indicate that the energy consumption and cache hit ratio of the proposed scheme are more effective than those of the two strategies compared: ABR-LRU and ABR-NonT. Table 4 presents recent studies focusing on energy efficiency in caching systems.

3.4. Latency

The time interval between the file request by UE and file delivery is referred to as *latency*. As application usage has increased, prompting higher QoE demands, the response time has become even more critical. It can be argued that latency is the most important metric for evaluating system performance because it is the foremost item that users expect to be perfect. This is why latency is always an important factor in technology transitions, such as from 4G to 5G. According to digital trends [126], the average latency of 4G is approximately 50 ms, while this number drops by a factor of five, to only 10 ms on average for 5G. In perfect scenarios, the latency can even reach 1 ms. In addition, the waiting time for data retrieval is money wasted and productivity lost. In many cases, delays translate into frustrated users, leading to plummeting user satisfaction. The network latency consists of three procedures: *i) processing delay:* the time routers take to process the requested data, *ii) queuing delay:* the routing queue time of data, and *iii) propagation delay:* the time for data to be delivered to users. The use of latency-aware algorithms has reduced the workload in communication networks while improving QoS [90, 127].

A belief propagation [128]-based algorithm for decentralized and collaborative caching is derived to solve the cache placement problem to reduce average download delay, subject to the storage capacity of BSs [115]. Each BS iteratively computes and exchanges belief information on the local caching method with surrounding BSs, collaborating on file transfers to their shared UE. In [67], a primal-dual decomposition approach was utilized to break down the problem and create an efficient content caching and delivery system in heterogeneous cellular networks to reduce average download time. Al-Turjman *et al.* [119] proposed a content demand ellipse (CDE) framework to create an SBS placement algorithm to alleviate IoT in ICN, determining the least number of SBSs in high-demand regions (hotspots), so as to maintain an upper bound for the UE delay, offload traffic from MBSs to SBSs, and minimize the global delivery cost. An IoT-specific integer linear problem (ILP) redistributes static/mobile SBSs to better release overhead from MBSs for increased content accessibility; a traffic analyzer based on CDE is used to address the problem. In [120], Fan *et al.* introduced a popularity and gain-based caching system to improve the hit ratio and minimize the UE request delay by requiring fewer hops for data transfer. In [34], to maximize overall income and spectral efficiency of the network, Wang *et al.* formulated the computation offloading choice, resource allocation, and content caching technique as an optimization issue. The problem was then transformed into a convex problem, decomposed, and the alternating direction technique of multipliers, which is influenced by the primal-dual interior-point approach, was used to solve the problem quickly and realistically. [129]. This decentralized approach converges rapidly after a few rounds, reducing computation complexity while providing comparable performance in terms of computing resources and spectrum allocation to the corresponding centralized algorithm.

Regarding caching schemes, Dai *et al.* [121] created two caching techniques that are fundamentally suited to CRAN and increased the transmission latency and hit ratio. The first considers UE mobility, while the second uses a Markov technique to calculate the video segment popularity. To realize these two schemes, Dai *et al.* constructed a CCU that contained a mobility estimator to obtain the probability of UE mobility, a bandwidth estimator to evaluate the status of the bandwidth of the BBU pool or individual RRH, a cache scheduler to accurately manage the process of data caching and the replacement process of CUs, and a UE video-segment-request table to record the index of video segments requested by the UE. In [122], Amer *et al.* described the latency per UE request in clusters, from a queuing standpoint. The original delay minimization issue was then equivalently changed to a particular form using a low-complexity greedy technique. This technique generates an effective cache placement policy to reduce the network average latency and outage probability. In [114], Lei *et al.* applied a deep learning approach to improve network performance in terms of latency and EE, through near-optimal and time-efficient solutions.

From a computation offloading perspective, Zhang *et al.* [123] jointly formulated the optimization task of computation offloading, content caching, spectrum, and computation resource

Table 5: Prime examples of latency-aware studies.

Ref.	Network model	Technical approach	Cache phase	Merits
[34]	HetNets	Primal-dual interior-point	Cache delivery	Tackle the joint problem of computation offloading decision, resource allocation, and content caching strategy to maximize total revenue.
[52]	MEC	Decomposition	Cache placement and replacement	Jointly utilize the advantages of reactive caching and proactive caching to perform a low-complexity heuristic caching strategy.
[67]	HetNets	Decomposition	Cache delivery	Simultaneously increase hit ratio and decrease average downloading latency.
[114]	HetNet	Deep learning	Cache placement	Because the computing cost is transferred to the DNN training phase, considerable complexity reduction in the delay-sensitive operating phase is possible.
[115]	MEC	Iterative	Cache placement	Present a low-complexity distributed algorithm which has comparable performance to that of the centralized greedy algorithm.
[116]	HetNet	Iterative	Cache placement	Propose a polynomial time algorithm to enable dynamic caching placement in HetNet.
[117]	Mobile networks	Suboptimal algorithm	Cache delivery	Get a lower-complexity distributed suboptimal method than branch-and-bound method.
[118]	HetNets	Stochastic geometry	Cache delivery	Apply stochastic geometry theory to have a better understanding of average network performance under fluctuating network node counts and placements.
[119]	ICN	Integer linear program (ILP) optimization	Cache placement	Propose a network that can be trained to adaptively construct an SBS placement algorithm with AI techniques.
[120]	ICN	Substitution method	Cache placement and delivery	Refine the popularity level of chunks to provide fine-grained caching control.
[121]	CRAN	MDP	Cache placement and replacement	Construct a CCU that fully utilizes the computational resources to solve the optimization problems rapidly.
[122]	MEC	Queue theory	Cache placement	Apply the analysis of queue theory to an inter-cluster D2D caching scheme.
[123]	MEC	Iterative	Cache placement	Propose a low-complexity iterative algorithm to minimize latency.
[124]	Mobile networks	AI	All phases	Survey on AI approaches to optimize latency in 5G and beyond mobile networks.
[125]	ICVN	Named data networking (NDN)	Cache Delivery & Replacement	RSUC and ReA are proposed for cache updating and content delivery for roadside devices design to provide up-to-date information with reduced latency.

allocation to minimize the latency of all computation tasks. Using the modified generalized Benders decomposition approach, the optimal solution is achieved in polynomial computing complexity time In [116], by loosening and dual decomposing the problem of limited resources and QoS, Liang *et al.* obtained the best solution to simultaneously offer proactive caching and bandwidth provisioning. The results show that the average delay of the system is reduced and the per-UE hit ratio is enhanced, which intuitively satisfies the UE QoS. Jiang *et al.* proposed a decentralized algorithm to distribute data content into clusters of SBSs based on the content state. When a content request, editorial change, or new arrival occurs, the algorithm is updated. By sharing the content state on a regular basis, these clusters generate a global state list. This information is then used to compute the optimal content distribution solution. In [52], Tran *et al.* suggested a low-complexity heuristic caching policy that combines a proactive cache placement method with a reactive cache replacement technique to provide at least one-half of the ideal value. In addition, they presented an online cache-aware request scheduling method that meets a formal competitive performance condition, while allowing the regulation of the content download pace and content access delay in a flexible manner. In [117], the authors initiated a low-complexity delay-oriented decentralized caching strategy for mobile networks to minimize the expected total latency of accessing the demanded content. This method can be operated in each BS, greatly reducing the signaling overhead among them. Li *et al.* [118] created decentralized caching optimization techniques via belief propagation (BP) by caching data into SBSs to minimize downloading latency, based on network structure. Furthermore, a fixed point in the proposed BP algorithm is demonstrated to exist, and in some situations, the BP algorithm is capable of converging to this fixed point. Within a limited margin, the suggested decentralized BP algorithm approaches the optimal performance of the exhaustive search. Additionally, a heuristic BP technique is developed to reduce communication complexity.

Recent studies on improving caching latency include the survey article of [124], which presents the implementation of ultra-reliable low-latency communications (URLLC) in edge caching using AI approaches, particularly deep learning (DL), deep reinforcement learning (DRL), and federated learning (FL). It was found that FL edge caching had the best performance compared to the others. In another study by [125], two approaches for cache updating and content delivery at roadside devices are designed to provide up-to-date information with reduced latency. In severely loaded ICVNs, roadside unit-centric (RSUC) and request adaptive (ReA) schemes can reduce service latency by up to 80% while ensuring content freshness. Surprisingly, the average age of information (AoI)-latency tradeoff is not always present, and frequent cache updates can affect both performance and reliability. Recent studies on latency awareness and minimization in caching systems are summarized in Table 5.

3.5. Spectral Efficiency

The rate of data delivered over a given bandwidth or spectrum band, in one second is characterized as spectral efficiency. The spectral efficiency has units of bps (bit per second)/Hz. Because of the shortage of available spectrum resources, caching

Table 6: Prime examples of spectral efficiency studies.

Ref.	Network model	Technical approach	Management model	Cache phase	Merits
[40]	Social networks	Optimization	Decentralized	Cache placement	Examine content delivery in wireless social network evolution scenarios.
[74]	CCN	Approximation methods	Centralized	Cache delivery	Propose a coded caching algorithm that maximize possible global coding gain, while respecting delay constraints of video streaming services.
[130]	ICN	Integer programming	Centralized	Cache placement	Propose a replication approach that scales with the expansion of the VoD service and the order of magnitude speedup, as well as a simple strategy for predicting demand for a new movie.
[131]	HetNets	Information theoretic caching	Centralized	Cache delivery	Investigate an arbitrary population distribution to derive a lower bound that takes into consideration all content to enhance the advantages of coded caching.
[132]	ICN	Deep reinforcement learning	Centralized	Cache placement	Adopt deep RL to process the CSI dataset from UEs, and then design the optimal policy for UE selection.
[133]	CRAN	Convex optimization	Decentralized	Caching strategy	Incorporate multi-cast beam-forming and content-centric BS clustering.
[134]	Wireless networks	Random caching	Decentralized	Cache delivery	In networks with user mobility, random caching methods are more realistic than deterministic ones.
[135]	N/A	Approximation methods	Centralized	Cache policy	Attains an approximate optimal memory-rate trade-off when hierarchically dividing the storage into levels.
[136]	D2D networks	D2D caching	Decentralized	Cache policy	D2D caching networks may convert memory into bandwidth (i.e., doubling the on-board cache memory on the UEs doubly increases UE throughput).
[137]	HetNets	Iterative	Centralized	Cache delivery	Considers the joint transmission scheduling and rate allocation problem of SVC streaming over cache-enabled HetNets.
[138]	HetNets	Integer programming	Centralized	Cache delivery	Find the set of layer caching indicators to optimally select the versions of the video file to be cached at SBSs.
[139]	MEC	Lyapunov optimization	Centralized	Cache placement	Initiate a dynamic approach for VNs to serve ABR video streaming services.

algorithms should preclude predictable content requests and consider spectral efficiency as a vital requirement. Spectral efficiency is used to evaluate the spectrum utilization of a cellular system. The higher the efficiency, the more the users that can be accommodated. The network throughput largely depends on the interference level and increases with the number of active users; therefore, expedient measures are required for network improvement [111, 127].

Qin *et al.* [40] found that users with strong social relations tend to request the same content; thus, content popularity is sharply concentrated and increases the content delivery rate. They proposed a routing scheme based on content popularity, formulated the maximum delivery rate problem, presented the evolution of users and content, and addressed optimal cache placement. After validating the theoretical results using a real data set [140], they demonstrated that the content delivery rate significantly improved with their routing scheme. In [141], an analytical framework was constructed based on optimal control theory and dynamic programming to design a cache replacement algorithm to best minimize server bandwidth cost. The authors in [142] used transfer learning (TL) to predict content popularity, with the most popular materials being proactively kept at the SBSs until their storage capacity was depleted. This new caching procedure showed higher user satisfaction as well as backhaul offloading gains in the case of sparse data and cold start problems. Nevertheless, since each SBS caches the most popular content individually, the same content is likely to be cached by many SBSs, culminating in duplicated caching and low caching performance. Tao *et al.* [133] formulated a sparse multi-cast beam-forming (SBF) problem for each multicast group, minimizing the weighted sum of backhaul cost and transmit power while adhering to QoS constraints resulting from UEs served by the same cluster of BSs requesting the same content. For simplicity, Tao *et al.* divided the original problem into two categories. The first category is the optimization of the content-centric BS clustering and is solved by approximating to a smoothed l0-norm problem, using convex-concave procedure based algorithms [143] to find an effective solution. The second is the effect of different caching strategies on the overall performance of CRAN and is addressed by sparse beam-forming algorithms.

Considering the cache content, Christopoulos *et al.* [144] derived which cache contents should either be broadcast or unicast to evaluate the gain in terms of overall spectrum efficiency and define the best content popularity threshold analytically based on an obvious cost function. To this end, cooperative multi-point joint processing techniques were used within an analytic framework to establish the content popularity threshold. In [16], Hou *et al.* proposed a decentralized caching system based on the MEC architecture, with the goal of reducing transmission costs while enhancing QoS. In this study, a learning-based technique was used to increase the accuracy of content popularity prediction. Golrezael *et al.* [134] investigated a cache-enabled wireless network that is resilient to UE

mobility by considering the deterministic caching strategy for a centralized version and random caching strategy for distributed UE. The system utilizes D2D communication links with high-frequency reuse, among UEs, to create large virtual caches, in which file duplication is avoided as much as possible. The incremental spectral efficiency obtained in this case, is on the order of one to two magnitude.

Ji *et al.* [136] showed that the integration of D2D spectrum reuse and caching at the UE, yields a D2D network in which any per-UE throughput is independent of the number of users and increases proportionally with the cache capacity of the UE. This means that caching in the UE can increase throughput by orders of magnitude without requiring a new bandwidth. In the proposed system, the optimal throughput-outage trade-off is achievable in terms of strict scaling rules for all system parameter scaling regimes. In [130], Applegate *et al.* jointly considered constraints of storage, link bandwidth, and content popularity as a mixed-integer program (MIP), and employed a Lagrangian relaxation-based decomposition technique combined with integer rounding to find a near-optimal solution with an order of magnitude speedup. The MIP-based approach obtains the number of copies required for each video where each copy is cached to save half of the total network bandwidth consumption, compared to LRU or LFU cache replacement policies. In [74], Niesen *et al.* coded multicasting for delay-sensitive services (with a strict deadline, e.g., video streaming), and suggested a computationally efficient content delivery method that makes use of coding possibilities inside a particular coded caching scheme.

To save the valuable bandwidth of the server, Niesen *et al.* merged requests as necessary to reduce the number of coded multicast packet broadcasts. In [131], Zhang *et al.* defined an arbitrary popularity distribution, which groups the most popular contents and less popular contents separately. They further suggest a new information-theoretical lower bound on any coded caching strategy anticipated transmission rate, and demonstrate that a simple coded caching technique achieves the predicted transmission rate. The resulting transmission rate was at most a constant factor, which is independent of the popularity distribution, away from the lower bound. In [135], Karamchandani *et al.* proposed a content caching and delivery technique for a two-level hierarchical coded caching network. This scheme uses two basic approaches. The first provides possibilities for coded multi-casting at each layer [27], whereas the second searches for coded multi-casting opportunities across several levels. The proposed caching technique achieves an approximately optimal memory-rate trade-off by finding the appropriate combination of these two types of programmed caching options. Both layers can simultaneously operate at an essentially minimal rate. In [137], Zhan *et al.* focused on transmission delivery for scalable video coding (SVC) [145], scheduling the optimal transmission from SBSs to UE and allocating the SVC video transmission rate for different UEs to maximize the total transmission rate while satisfying QoE requirements. Zhan *et al.* derived a heuristic solution based on an iterative algorithm by relaxing the constraints.

Within a constrained environment, Zhang *et al.* [138] derived a near-optimal heuristic algorithm to obtain the layer caching indicators to optimally select which versions of the video file should cache locally at SBSs. Under the cache size constraint of each SBS, the total amount of data traffic that can be sourced from the local cache of the SBSs is maximized. In [150], Sun *et al.* solved the joint optimization problem of maximizing the revenue of CPs and backhaul usage by adopting a Lyapunov optimization framework, and then recommended a caching policy based on projecting the total number of UE requests. In [132], He *et al.* used deep RL to find the best interference alignment (IA) UE selection policy in cache-enabled opportunistic IA [151] wireless networks with time-varying channel coefficients. A central scheduler collects channel state information (CSI) from each UE. The integral system information is then sent to the deep network to identify the optimum policy for UE selection. In [139], Guo *et al.* proposed a time-scale dynamic caching scheme that works at both the application layer and the physical layer as well as for BSs in vehicular networks (VNs) for ABR streaming without prior knowledge of channel statistics. The Lyapunov optimization technique was employed to obtain the optimal decisions of video quality adaptation, video cache placement, and video data transmission. Table 6 lists recent studies targeting spectrum efficiency in caching systems.

3.6. Service Availability

Service availability is built based on the minimum level of QoS. While UEs cannot experience the remarkable latency and the quality of content, they are affordable at some threshold [16, 127]. In typical contexts, QoS directly reflects UE satisfaction. Ji *et al.* [152] defined the outage-throughput trade-off problem in D2D networks and suggested using a combination of caching and coded multicast transmission to simultaneously satisfy all UE requests simultaneously. The outage-throughput pair is achieved if the cache placement and transmission strategy meet both the minimal per-user average throughput via active D2D lines and the average outage probability. Hou *et al.* [16] reduced the transmission cost while improving the user QoS. To obtain optimal video streaming quality, Qiao *et al.* [146] formulated a dynamic proactive cache memory allocation problem for UEs traveling across cells as an MDP. To provide a universal and practical solution to the MDP, an approximated cell-by-cell decomposition approach was proposed. The video quality consistency (QoS) is maintained at a high level; the delay effect is eliminated; and video stalling, which occurs when the cached video material in the BS is insufficient to sustain video quality for a set period, is reduced. In [147], the authors optimized the effectively pleasurable video quality of all UEs while avoiding playback delay in video streaming services and offered an SBS association method for UEs to pick the right quality to interact with the surrounding UEs, with cached requested contents. In [127], Huang *et al.* proposed a D2D-assisted VR video placement algorithm that was designed based on user residence time, UE interest, and SBS popularity to maximize the QoE gain of all UEs. The SBSs communicate and pre-cache the videos because of the estimated probability of a UE entering each SBS. Distributing Internet services on cloud servers on an ongoing basis is expensive and consumes a lot of

Table 7: Prime examples of service availability studies.

Ref.	Network model	Technical approach	Management model	Cache phase	Merits
[14]	Mobile Edge Network	ILP & Randomized rounding & Game Theory	Both	Cache placement	Minimizing cost when caching Internet services at edge clouds and studying whether cooperative or individual content sharing would be optimal.
[127]	D2D networks	Optimization	Centralized	Cache placement	Develop a VR video caching algorithm to ensure high QoE gain for UEs.
[146]	5G networks	Decomposition method	Centralized	Cache placement	Propose a memory allocation method to effectively maintain non-real time video streaming quality when the UE and BS have very short connection time for data transmission.
[147]	D2D networks	Lyapunov optimization	Decentralized	Cache placement	Propose a UE association algorithm for file delivery.
[148]	Two-tier cellular network	SDAS	Centralized	Cache replacement	Work on information-centric coverage probability performance, bring novel insights into the architecture of a two-tier cellular network.
[149]	Large-scale WiFi	Optimization	N/A	Cache placement & replacement	LEAD scheme was proposed to maximize long-term gain for edge caching in large-scale WiFi networks despite the heterogeneity of traffic.

resources in 5G and beyond networks. Caching services that are regularly utilized in the edge cloud are an alternative method. The work in [14] attempted to solve the problem of minimizing costs in two different scenarios. When there is no content sharing among ISPs, an integer linear program (ILP) and a randomized rounding algorithm are used. On the other hand, a game-theoretical mechanism is used when ISPs share content. Consequently, by allowing ISPs to cooperate and share data, the system performance can be increased, and the cost can be greatly reduced. Lyu *et al.* [149] implemented a large-scale WiFi system with 8,000 access points (APs), serving approximately 40,000 users actively for two months. The problem of deploying edge caching in these types of extensive WiFi scales is the heterogeneity of the traffic load. The goal is to maximize the long-term caching gain by proposing a *large-scale WiFi edge cache deployment* (LEAD) caching strategy. Table 7 summarizes recent studies on caching service availability.

3.7. Metrics Trade-off

The constraint of processing capabilities on edge servers has undoubtedly impacted MEC performance. The amount of cachable content grows in tandem with the rise in data traffic. Meanwhile, user requirements are getting more demanding over time, requiring MEC deployments to use more energy in order to remain operational and maintain ideal features. To maintain the quality of services under such resource limits, a feasible solution is to efficiently trade-off the performance of metrics and the quantity of cache contents in order to equally divide resources and assure the flow of energy required to maintain the quality of MEC services. One of the most straightforward approaches is to trade-off monetary expenses for better MEC operation, which means users must pay to get additional computing resources. This method has been addressed in [153], this approach has been discussed in which the QoE optimization can be achieved by trade-off the latency and the costs. In practise, however, most users are unwilling to spend extra to optimize a strategy that they may have never heard of. Another promising trade-off is to reduce the amount of cached contents at MEC servers to save computing resources for optimizing MEC metrics. By having suitable caching strategies, as have been previously mentioned in 2.4, edge servers can select contents which have high probability of being requested and cache them, rather than caching a variety of different contents.

A consistent and realistic trade-off approach is to consider each characteristic of MEC. For example, in [154], the authors have considered caching both popular and unpopular contents to increase the cache hit ratio, meaning trading off the storage efficiency for hit rate. Similarly, Asheralieva and Niyato [155] proposed an integrated contract theory-Lyapunov optimization content sharing scheme which trade-off the network cost with the total stability of edge caching system. In particular, by increasing network cost, the content delivery delay which was denoted as Lyapunov drift can be minimal, hence achieve more stable state for caching system. In blockchain-enabled MEC systems, for such energy expensive framework, Feng [156] has trade-off the performance between MEC and blockchain by simultaneously considering user affiliation, bit rate distribution, block generator scheduling, and computing resource allocation. The simulation results demonstrated that the energy required for the entire procedure could be lowered while still ensuring acceptable MEC and blockchain performance. Most trade-off approaches can be treated as optimization problems and addressed using optimization tools and algorithms. Nonetheless, scientific paper nowadays can only maximise particular features rather than considering the complete system as a whole, therefore the trade-off of all MEC properties has yet to be examined.

3.8. Summary and Discussion

In this section, six different characteristics for caching performance evaluation were analyzed: cache hit ratio, storage capacity, energy efficiency, latency, spectral efficiency, and service availability. For each metric, the network model, technical approach, cache phases, and merits were specified. Many studies have been conducted to maximize the cache hit ratio. Al-

though popularity-based caching is the most prevalent option, it is difficult to precisely estimate the popular and appropriate content for each user. As a result, having an effective caching method can help the cache achieve high probability prediction. Cache storage capacity is typically restricted, and in most cases, the solution would not be to extend the storage, but rather to effectively select the contents to cache, or to cooperatively share cache contents across edge nodes. As a result, the energy required to calculate and operate the cache system can also be reduced. Energy optimization is crucial in 5G and beyond technology development, as edge clouds may now be utilized for caching, computing, and pre-processing data on near end-user devices in addition to storing. Another aspect that may lead to insufficient energy usage is the execution of the optimization algorithms. Furthermore, when the number of users and requested items increase rapidly, the processing time may be delayed, resulting in lengthy latency. Latency can occur at any point in the caching process, not only while delivering content to users. Latency is usually the first metric used by users to evaluate system performance. High latency can negatively affect the QoE of users, leading to plummeting user satisfaction. Spectrum efficiency is defined as the bandwidth of edge caching servers. Optimizing the spectral efficiency can boost user accommodation. ISPs are working to expand spectral efficiency when deploying 5G and 6G networks to provide reliable and cost-effective service coverage. Massive MIMO is an example of spectrum optimization. A system with high antenna count that is resistant to frequency interference, improves the accuracy of beamforming towards the users. Next, the service availability of edge caching, is built based on the minimum level of QoS to satisfy QoE. Last but not least, a discussion on trading computing resources was mentioned as a viable solution for resource limits at edge servers. Despite various attempts to tradeoff MEC aspects, there is a need for a more complete and optimum approach that takes into account all MEC system characteristics.

4. Caching Models and Techniques

4.1. Information Theoretic Caching

While most other techniques treat content placement as a separate issue with unicast and multicast data transmission, information-theoretic approaches exploit the advantages of cooperative placement and coded transmission [70]. In the aforementioned work of Maddah *et al.* [72], an information theoretic formulation for the caching problem was introduced in coded caching consisting of a placement phase and delivery phases. The purpose of coded caching is to reduce the peak rate of the shared bottleneck link by adjusting the placement and delivery phases concurrently to satisfy many UE requests with a single coded multicast broadcast. In [157], the caching problem was formulated as a distributed source-coding problem with side information. Wang *et al.* [157] treated the requested data from UEs as a function of the request of the entire data. For the single-user case, they established a single-letter assumption of the optimal rate region and derived closed-form expressions for the special case of uniformly distributed UE requests. Insights gained from the single-user case were applied to the three two-user cases, and a single letter expression of the optimal rate region was obtained accordingly. In [158], an information-theoretical approach was introduced to balance the trade-off between load and user privacy in caching systems, particularly in the caching placement phase. Using the Pareto optimal method, privacy can be maximized with an optimal traffic load.

4.2. Game Theoretic Caching

Game theory is a mathematical modeling approach that attempts to simulate a system in which players strive to maximize their benefits. This model may fit MEC networks because BSs, UEs, or even MNOs can be considered players. In game theory, the algorithms are classified into two sub-groups: hierarchical (Stackerlberg) games and matching games.

Hierarchical (Stackerlberg) games: In [159], Hu *et al.* recommended using game theoretic approaches for wireless proactive caching with four typical scenarios in terms of problem-solution pairs, including SBS caching and auction games, roadside unit caching and contract games, D2D caching for mobile users and coalition game, V2V caching and evolutionary game. The authors define the vital requirements of each scenario and match them with the game-theoretic approach capabilities. The Stackelberg game is a strategy game in which a leader and numerous followers compete for limited resources [160]. The leader advances first, followed by the following: In [161], Su *et al.* developed an edge-caching framework to cache layered videos. Initially, they model the interaction between the UE and BS by a Stackelberg game and the competition among the UEs by a non-decentralized game. Using the backward induction approach, the ideal strategy for each participant is determined, resulting in a greater hit ratio and shorter latency of video content delivery. In [162], Li *et al.* investigated a commercialized small-cell caching system consisting of a video retailer and multiple CSs and proposed a Stackelberg game-theoretic framework to jointly maximize the average profit of video retailers as they lease their videos to the CSs and the individual CSs as they rent popular videos from the video retailers and cache them into their SBSs. The Stackelberg equilibrium (SE) is investigated by solving a game-theoretic optimization problem, indicating effectiveness in pricing, hit rate ratio, and backhaul transmission. Alioua *et al.* [47] also used a Stackelberg game to solve the problem of V2V caching in vehicular networks.

Matching Games: The matching games provide solid mathematical tools suitable for studying the considered problem[163] and are divided into three sub-categories: One-to-One, Many-to-One, and Many-to-Many. In [164], Hamidouche *et al.* developed a many-to-many matching solution between the two sets of SPSs and SBSs while considering the restricted capacity of backhaul links, and then proposed a decentralized algorithm to prove that a pairwise stable outcome can be attained. Each SPS that hosts videos is associated with a set of SBSs so that relevant videos can be cached. In contrast, the SBSs are associated with a set of SPSs and, indirectly, with a collection of movies stored

in those SPSs. Afterward, they compete to see which files to be cached, at which SBS. The set of videos assigned to each SBSs is strategically decided. The algorithm boosts the number of satisfied requests (QoS) and significantly reduces the latency of UEs.

4.3. Machine Learning

Online Learning: Online learning requires low-complexity algorithms for future 5G networks, which have several latency-sensitive applications. Recent research has examined the effect of online caching on improving the performance of cache-enabled networks [165, 166]. To deal with unpredictable and changeable content popularity among users, Muller *et al.* [165] proposed a context-aware proactive caching method based on contextual multi-armed bandits for wireless caching entities. The algorithm changes the cached material by watching the context information of the connected users regularly and then observes the cache hits. In short, this algorithm learns the popularity of context-specific material on the Internet and increases the expected hit ratio by at least 14%, compared to popular cache placement algorithms, using a real-world dataset. Li *et al.* [167] proposed a trend-caching algorithm to forecast the trend (i.e., future popularity) of video content to make proper cache replacement decisions, while the improvement in hit rate ratio possibly exceeds 40%. This trend-caching algorithm learns the interaction between the trend of content and the context in which the content is requested. A training phase and a priori knowledge of popularity distribution are not required.

Using processing capabilities and memory storage of smartphones, network operators may proactively serve the expected peak-hour demands during off-peak periods [22]. When the proactive network fulfills user requests ahead of their deadlines, the related data are saved in the user device, and when the request is placed, the information is retrieved directly from the cache memory rather than querying the wireless network. Novel machine learning algorithms should be developed for this purpose in order to establish optimal trade-offs between predictions that result in retrieval of information users never sought and requests that were not anticipated in a timely manner. Machine learning techniques can provide effective caching in 5G cellular networks, allowing a CCU to learn, track, and even react to the popularity of reusable content [168]. Bacstuug *et al.* [168] tightened the connections of *big data* by introducing a proactive caching architecture in which statistical machine learning technologies are used to estimate content popularity, as the first attempt to shed light on the enormous potential of big data. Bacstuug *et al.* [45] presented an architecture to simultaneously handle the computation and implementation of cache policy content prediction algorithms at BSs to tackle the complex problem of content popularity estimation tied to the spatio-temporal behavior of UEs. As a result, depending on storage capacity, multiple caching advantages in terms of QoS and spectral efficiency become conceivable.

Transparent Learning: Transparent computing (TC) [169] was first proposed in 2004. The core idea of TC is that all data and software, including operating systems, applications, and UE data, are stored on the server side, whereas data computing is performed on the client side. Zhang *et al.* [169] introduced the superiority of MEC into TC through transparent learning (TL) technique, which performs data training on the server and stores the test models on the client side. The test models are updated using incremental training. TL consists of three main parts: a transparent client, transparent server, and edge nodes.

Deep Learning: With the heterogeneity of network traffic and high randomness in content popularity and user preferences, deep learning is an acceptable approach for solving these hard optimization problems. The survey conducted by Wang and Friderikos [170] analyzed and compared in detail the state-of-the-art deep learning approach used in recent research for data edge caching, including supervised learning, unsupervised learning, and reinforcement learning. They also highlighted some difficulties in using deep learning, such as deployment cost, dimension caching, and augmented reality (AR) applications caching. In [171], the authors proposed a deep learning method to improve the QoE for users and decrease the workload of networks. In particular, a proactive sequence-aware content caching strategy (PSAC), which consists of two frameworks, a convolutional neural network-based PSAC-gen and an attention mechanism PSAC-seq. Although the proposed schemes can achieve the desired results, they have not proved their effectiveness in real-world scenarios, and latency is not guaranteed.

Deep learning also has great applicability in the autonomous car field; therefore, the integration of edge caching and self-driving cars can receive many benefits by leveraging deep learning. Ndikumana *et al.* [172] proposed infotainment caching in an autonomous car, where the caching system caches content by analyzing passenger characteristics with minimum delay. The caching process was divided into four parts: predicting content, retrieving, caching, and delivering content. Deep learning was used during the first phase of predicting the content of the cache. By using this scheme, 97.82% of the contents were successfully cached with minimum latency.

Transfer Learning: Unlike typical machine learning algorithms, which learn each task ground up, *transfer learning* (TL) approaches, transfer knowledge from prior relevant tasks to a target goal using fewer high-quality training data, to resolve future issues more rapidly and to come up with better answers. In [16], Hou *et al.* introduced a transfer learning-based decentralized proactive caching mechanism (LECC) to measure content popularity and solve the optimal caching problem using a greedy method. The proposed algorithm outperforms well-known caching policies such as LRU, randomized replacement (RR), and centralized learning-based caching strategies for cache hit rate, average content delivery latency (ADL), and transmission cost. Furthermore, the performance benefits of LECC are comparable to those of the popularity-aware greedy strategy (GT) based on real content popularity, demonstrating the efficacy of incorporating intelligent assessment of content popularity through the use of TL-based techniques. In [23], Bharath *et al.* demonstrated the connection among the popularity distributions of UEs, SBSs, and BSs under independent

Poisson point processes (PPPs) and proved the lower bound on the training time of a TL-based approach to achieve a specific level of estimated accuracy ϵ. This lower bound is inversely proportional to λ_u and λ_r, which indicates the densities of PPPs with respect to the UE and requests. It also scales as B^2, where B is the file size. The fundamental concept for 5G caching stems from the fact that only a small proportion of requests are answered during a caching period. This, along with the small dimensions of cells, can make it difficult for SBSs to correctly estimate the underlying content popularity. To address this issue, the works of [173, 23] advocate a transfer-learning strategy that leverages past knowledge gained from a proxy domain (e.g., social networks) to enhance time-invariant popularity profile estimations. Bastug *et al.* [174] used transfer learning to alleviate data sparsity in the information on social networks. Leconte *et al.* proposed a threshold policy named age-based threshold for the timely exploitation of time-varying popularity to improve the hit ratio rate. In addition, this study proposes a system that incorporates global learning and local caching to minimize wireless content access latency and minimize incurred traffic.

Reinforcement Learning: Through interactions with the network environment, the optimum stochastic strategy is learned using reinforcement learning (RL) approaches. When a user interacts with a cache, most of the time the cache system does not have prior information on users, such as network condition, location, and content preferences. Caching can effectively cache after several iterations of learning the behavior of users. For example, when a new account is created, YouTube can only provide the user with some common popular videos, which may not be what the user wants to watch. After the user has searched and watched some particular videos, the caching system learns the genre of those videos, and can use them to cache more relevant content in the future. In [175], Wei *et al.* used an actor-critic [176] deep reinforcement learning method to minimize the transmission delay of all contents in a HetNet with the SBSs acting as relays. In this case, to assist the actor in adjusting the stochastic policy, the DNN is employed as a value function approximation. These solutions assume that the service of the current user requests can be completed before the next request arrives, and hence no buffer is required to store user requests. During peak hours, however, the user request rate is high. Many user requests will be dropped if there are no queues or buffers to hold incoming user requests when the system is busy. Thus, the aforementioned scheduling algorithms may not be adequate for peak-hour schedules. In [177], secure edge caching is proposed to address the vulnerabilities of edge caching systems to cyberattacks as well as cost insufficiency. To resolve the selfish reaction of caching devices that incur a cost burden, a Stackelberg game scheme was used to encourage content-sharing cooperation between the ISP and cache device. As for the reinforcement approach, Q-learning was conducted to solve the security issue.

Deep reinforcement learning (DRL) is an advanced combination of reinforcement and DL techniques that approximates the Q value-action function using a deep Q network [178]. It can be said that with most of the hardness characteristics of optimization problems in edge caching, DRL is the most prominent, effective, and most common approach to solving these disputes. Google Deepmind uses this strategy in various games and achieves acceptable results [178, 179]. Wang *et al.* [180] proposed a federated DRL-based cooperative edge caching scheme to tackle the computational resource shortage in the edge caching process. As mentioned in Section 2.4.2, FL is the integration of one centralized caching decision-maker, which applies a common caching strategy to all decentralized cache systems called ”agents”. In this study, these agents are base stations. FADE uses the information of the first cache from the BSs as input and feeds it to the decision-maker. Once the optimal cache strategy is generated, it is sent back, and applied to all BS caches. Having a federated scheme may not provide the best caching strategy for each BS; however, it is optimal for the global system. This would help the decision-maker use less computational resources for decision learning, instead of having to learn each BS. Offloading backhaul traffic, latency, and performance insufficiency decrease, and the cache hit ratio improves.

In vehicular edge caching, [181] also used a DRL approach and blockchain to perform secure and optimal caching in vehicles. In particular, the authors constructed a riskless, high-secure caching for BSs, then leveraged DRL and the geographical location of the vehicle to generate an optimal caching scheme. Finally, the proof of utility (PoU), a novel block verifier selection mechanism was developed to speed up the block verification process. Integrated DQL was introduced in [182] to support MEC at vehicular network. The rise of intelligent connected vehicles have created an enormous task offloading burden for MEC servers. The proposed idea was to develop a distributed computation offloading scheme to efficiently utilize resources from all connected vehicles and minimize the execution time for offloading tasks. It can be said that the scheme is similar to the above mentioned FL approaches, since it jointly consider the entire nodes for optimizing overall performance.

4.4. Summary and Discussion

Edge caching methods have been divided into three main techniques: information theoretic caching, game theoretic caching, and machine learning. There is a paucity of contemporary studies utilizing information theory. Information theory is typically used to solve optimization problems in content placement mostly because it is outdated and the implementation is complex making it less prevalent than other methods. The game theoretic technique emulates optimization issues as a game strategy to optimize the result rewards and increase the caching strategy learning process. The two most common game theories are the Stackerlberg game and the matching game. Game theory may also be utilized to boost cache competition while reducing player selfishness to obtain the greatest effective shared benefit for the global system. Finally, machine learning has garnered much attention from today’s scholars and is regarded as the simplest and most successful method for solving optimization issues. This is an inevitable trend because machine learning offers easy, cost-effective, and effortless solutions. On the flip side, future solutions will heavily rely on machine learning,

unknowingly giving it data and inputs, and expecting an easy solution, rather than addressing the underlying problem.

Nevertheless, to be able to deploy ML and AI approaches such as DL and DRL onto MEC requires an intensive amount of energy sources from edge servers. This has raised the need for solutions to the energy efficiency problem. One traditional solution is to directly provide energy for BSs and edge servers through wired connectivity. Tethered or ground wired connection for terrestrial BSs has been used for its simplicity. Another alternative is to optimize the resource allocation capability by efficiently distributing energy for different tasks in order to decrease energy bias while guaranteeing overall system performance. The primary idea for this method is comparable to the trade-off technique, as previously discussed in Section 3.7, which means the resource allocation for edge servers was considered as optimization problem. Nevertheless, using this solution also has to consider utilizing ML, which may potentially resource costly. To this end, an idea that has recently been considered as possible and highly promising is the utilization of the energy supply of edge devices. Instead of heavily rely on the resource storage of edge servers, this attempt allows MEC systems to leverage the abundant resource from UEs. FL and split learning are some of the latest techniques which were formed based on this idea. The collaboration between edge servers and UES can even further enhance the security and privacy of user data.

5. Application Scenarios

5.1. Social Media Platforms

The rise of social media platforms such as Facebook, Instagram, and Twitter has created valuable opportunities to develop algorithms to exploit the social relationships among users. Social relationships are characterized by external influences, such as media and friends as well as user interactions and links. For example, a dynamic social network can comprise students, faculty, and staff of a university in which interactions between UEs are inferred from e-mail headers (timestamp, sender, and list of recipients) and are matched with personal attributes (status, gender, age, number of years in the community) [183, 184]. By leveraging the link between user data, their social interests, and their shared interests, the accuracy of forecasting future occurrences (i.e., user geographic placements, next visited cells, requested files) may be greatly increased. The new nodes attach to existing nodes with a probability proportional to the node degree [40]. The vast bulk of data flow from social networks will continue to impact the way people acquire information [40, 161]. Because UEs with strong social ties tend to seek the same material, the distribution of content popularity is highly concentrated [40]. In addition, social networks can evolve and stabilize the delivery rate despite the increase in network size.

The diverse content environment in social media platforms may be a burden on UE energy, resulting in a tremendous delay in data transmission. An efficient content-centric edge caching integrated with ML approaches is considered an optimized solution for the aforementioned issue. Aftab *et al.* [185] proposed a community-based clustering framework to predict the common interests of users and used a hybrid scheme with a combination of mini-batch K-means and DBSCAN for edge caching contents to address this problem. The results indicate that latency and energy inefficiency can be greatly reduced by caching content near end-users.

5.2. Economy

The edge caching technique has a significant impact from the economics perspective. By predicting and caching popular contents and storing them near end-users, it becomes possible to save on computational resources, thereby decreasing the cost. Gharaibeh *et al.* [166] studied the problem of content distribution in content-centric networking (CCN) and proposed an online CCN caching method which operates per basis and does not need precise content popularity to reduce the overall cost paid by the content provider (CP).The total cost is calculated as the sum of the caching expenses consisting of the payment to the ISP in return for caching its content items, retrieval charges, and the estimated cost of losing UEs to other CPs. Moreover, edge caching significantly reduces the latency of content delivery, thereby improving the QoE of users. By ensuring user satisfaction, ISPs can attract new users to subscribe to their services, thereby increasing their profits. In [186], QoE was enhanced through a deep reinforcement learning approach for big data architecture in edge caching, at a lower cost. However, the work did not consider working in a heterogeneous environment, privacy and security, or on-device caching.

5.3. Web Caching

The most prominent websites are experiencing significant congestion as a result of millions of requests every day, regardless of special events. Web caching [187] is a technique that may dramatically improve end-user web surfing while also saving bandwidth for service providers. In particular, a web cache is a temporary storing location for data material retrieved via the Internet. The most popular web browsers cache web pages that have been visited by UEs and enable access to already cached pages/content by simply clicking on the back button. At the network level, objects can be cached on proxy [188] and on web servers at the network edge, ISPs, regional and national Internet hubs, etc. Hasslinger *et al.* [189] investigated the effect of different item sizes on web cache speed as well as overhead and proposed a class of rank exchange caching strategies. As a result, for objects of varying sizes, the update effort was estimated to increase by a ratio of 2-3. The average update effort per request remained constant, while the update speed scaled to millions of requests per second. Another recent work on optimizing web caching performance [190] used Knapsack solutions for caching, considering Belady's optimum strategy for clairvoyant caching. They also compared the caching performance between the LRU and GreedyDual schemes and found that LRU was completely outperformed due to its inflexibility in complicated networks.

5.4. Internet of Things (IoT)

According to the Cisco Annual Report 2020 [1], the Internet of Things (IoT) has experienced an explosive exponential increase, whereby IoT devices are expected to reach 14.7 billion in 2023. This indicates that IoT technology has enormous growth potential, followed by technologies that it is integrated with, particularly edge caching. In brief, IoT is an interconnection among devices, computers, and systems that exchange data and information through the Internet. When integrated with edge caching, IoT data can be stored at the network edge, thereby reducing delays in data transmission using backhaul links [83]. It also increases energy efficiency and reduces cost by improving backhaul link utilization and minimizing retrieval time for IoT data. However, in practice, the limited cache storage of edge servers and nodes, in addition to communication overhead in terms of cache coordination and global management of the in-network storage resources, make the implementation of an efficient IoT-edge caching system complex. Given the discrete nature of IoT, there are also security and efficient resource utilization concerns. This issue is also stated in [191] where the offloading rate is maximized to improve the caching probability. First, they present a framework to formulate cache hit as a stochastic problem. Subsequently, an improved caching probability conversion (CPC) algorithm based on the Monte Carlo method is developed to resolve and optimize. Another study by Sheng *et al.* [192] also used DRL to enhance the energy efficiency and cache hit ratio in IoT systems. The advantage actor critic (A2C)-based algorithm is designed to maximize long-term energy savings while ignoring IoT data popularity characteristics. In [193] the problem of edge cache hit ratio is considered using IoT with Blockchain, a potential paradigm for decentralizing a single trustworthy entity conventional ledger, which is an emerging technology in cryptocurrencies and internet security. The Markov decision process is used to improve the node selection strategy of content deployment, QoS is increased; bandwidth waste is decreased; and the hit ratio is boosted. A disadvantage of works on IoT edge caching is that they make a perfect scenario assumption in the simulations, rather than considering the real-world.

5.5. Video Streaming

Video streaming services are expected to cover 74% of the total data traffic by the end of 2024 [194], becoming the most popular type of online entertainment service. The surge of video streaming service giants such as YouTube, Tiktok, and Twitch has allowed more users to access video from their devices. In addition to the development of mobile devices and the number of smartphone subscriptions skyrocketing over time, user expectations for video streaming have also increased. User behaviors change as network capabilities grow, and this is expected to further increase when 5G services become accessible. High-definition (HD) video streaming at 720p and 1080p resolutions is expanding, and the average resolution of a YouTube video is already up to 720p. Video services place a large demand on low latency, making resource allocation highly challenging [195]. Furthermore, at the time this survey was being conducted, Facebook announced that it would deploy *Metaverse* in virtual reality and augmented reality environments. This will lead to a revolution in the field of video streaming in 5G/6G with an explosion in video traffic. To meet the demands of video transmission while satisfying the QoE of users, edge caching has been considered as a feasible solution.

The survey in [196] provided a comprehensive overview of edge caching, as well as edge computing and communication in video streaming. The authors briefly explain the usage of MEC-integrated systems to exploit UE resources to provide network satisfaction in terms of QoS/QoE, data rates, latency, etc. They also state current challenges in video edge caching, such as storage limit, sustainability, security, and privacy. In practice, video caching is complicated because of the diversity of request patterns as well as its dynamic essence. In the aforementioned work of [83], a cooperative approach was introduced to reduce latency and cost. However, the similarity of requests among nearby edges may be quite dynamic and different, thereby limiting the benefits cooperative caching can bring. For this reason, Wang *et al.* [197] proposed the *MacoCache* scheme, a multi-agent deep reinforcement learning (MADRL)-based method to resolve these issues. In a real-life scenario, the latency and cost of the system can be reduced by 21% and 26%, respectively.

Adaptive bitrate (ABR) video streaming services are another solution for ensuring desirable video transmission. By analyzing the user maximum network condition and the highest adaptable bitrate, ABR requests, from the video server, suitable video chunks in which quality corresponds to the analyzed bitrate. Thus, users are guaranteed to receive the highest video quality and lowest latency possible, even with limited network connectivity and increased QoE. In [184], the problem of maximizing video bitrate was formulated as an integer linear program (ILP) and solved with an online iterative greedy-based adaptation (OIGA) algorithm. The simulation results show that the video bitrate can be maximized because of the flexibility of the algorithm in video popularity and retention rate adaptation. Zhang *et al.* [198] also attempted to use ABR to improve the QoE of users in a super-resolution edge-integrated system called VIdeo super-resolution and CAching (VISCA). One aspect that makes this work outstanding compared to others is that the framework was tested in a real-world environment whereby video quality increased by 28.2%–251.2% and re-buffering time dropped by 16.1%–95.6% for all scenarios considered, proving the functionality of the framework.

Virtual reality (VR) and augmented reality (AR) video streaming are also noteworthy. VR has a wide range of applications, from entertainment services such as video streaming, gaming, and social media platforms (i.e., Metaverse) to medical and military simulations. According to the Ericsson Mobility Report 2018 [194], for every five minutes of video streaming, approximately 12GB of VR with 1080HD resolution and 28 GB of 25 Mbps AR are generated. Old technologies may have held back the development of VR because of high requirements placed on bandwidth, latency, and computational resources. Nevertheless, with the deployment of 5G technology and beyond, a rise in VR is inevitable. In [199], the authors stated the problem of exceed-

ing the bandwidth requirements in VR video streaming, makes it difficult to deploy in a live-streaming setup. To resolve this problem, long short-term memory (LSTM) networks are used to predict the growth of content requests and to prefetch content to caches. Their solution when compared with other algorithms, such as least frequently used (LFU), least recently used (LRU), and first in first out (FIFO), outperformed the others in terms of delivered video quality, cache hit ratio, and backhaul link usage. Liu *et al.* [200] introduced a deep deterministic policy gradient (DDPG)-based framework to address the issue of simultaneous resource allocation and replica selection in blockchain-enabled fog radio access networks (F-RANs). As a result, less energy was consumed owing to efficient resource utilization and a good load trade-off.

5.6. Device-to-Device Communication

Device-to-device (D2D) communication is a feasible solution in assisting MEC to offload traffic [201]. D2D enables nearby users to communicate directly with each other, featuring high bandwidth efficiency, high data rates, and low delay [202]. On the other hand, because D2D is heavily dependent on users' willingness to share data and content, creating a trusted framework among users in which privacy and security are ensured is critical. Zhang *et al.* [201] proposed a consensus mechanism based on a partial practical Byzantine fault tolerance (pPBFT) protocol to guarantee the latency and security of the scheme. In addition, they formulated the caching placement and smart contract execution nodes (SCENE) selection as an MDP problem and solved it using the DRL approach. In the appropriate domain of low outage probability, the throughput of a D2D caching network performs similarly to that of coded multicasting [27], while the architecture is substantially simpler for actual implementation. This network throughput increases linearly with cache size and is proportional to the number of files, but the number of users has an impact on the result. Furthermore, the D2D caching network takes advantage of intensive spatial reuse, that is, repeating the same file several times in the network such that any user can find the desired content at a short distance with a high probability as a result of many concurrently active links being supported in the same time slot. Therefore, D2D caching networks are well suited to dealing with scenarios in which a small library of popular files is requested by a high number of UEs.

D2D and edge caching are also considered as solutions to avoid duplicate content downloading from the cloud. In [203], the authors evaluated the vulnerability in the cache-replacement phase of a real-world D2D edge caching policy (i.e., adaptability, privacy). A weighted distributed DQN model (WDDQN) was introduced to solve the aforementioned problem. In terms of the request hit and offload rates, the proposed approach outperforms the FIFO, LRU, LFU, and centralized DQN schemes, and it also has a faster convergence speed than the centralized DQN model. Another work [204] proposed distance-based and priority-class-based cooperative cache replacement algorithms to study the effect of video characteristics on system behavior, thereby improving energy consumption and increasing service capacity.

5.7. Summary and Discussion

Edge caching is critical in the deployment of 5G/6G technology because of its versatility and broad variety of applications. First, social networks profit immensely from edge caching because caching and presenting users with relevant material based on user preferences improves QoE and hence attracts more users to the platform. However, abusing user interests may potentially disclose personal data; therefore, privacy and security are important issues to address. The variety of materials on social media also results in delays and computing resource strain. Edge caching assists ISPs in saving a significant amount of money on service rollout and content delivery to subscribers. There is reduced computational stress on the system as a result of utilizing edge resources and storage, providing ISPs with additional opportunities to deploy services. The reduction in latency during the distribution phase also results in more clients expressing a more favorable opinion of the service increasing the likelihood of subscription, which results in increased profits for ISPs. Web caching is a clear illustration of this, in which favored websites are cached and prefetched to some extent to reduce user access time and provide additional services to consumers.

Another point worth considering is the use of edge caching in the IoT industry. Edge caching is regarded as the ultimate app, elevating both technologies to greater levels. As the number of IoT devices has increased, a large amount of data and information need to be stored, evaluated, and transported across the system. Caching at edge networks close to the end user minimizes retrieval time and the backbone network burden. Using edge computational resources, analyzing and computing data at the edge node help reduce the computational stress on the global system. Nonetheless, security and efficient utilization are problems that should be carefully examined throughout system operation when using IoT-edge caching. Because of the increase in video demand, edge caching in video streaming has emerged as a viable field of development. By providing users with their favorite videos and content, popularity-based caching may make full use of the possibilities in this sector. Finally, edge caching improves D2D communication by providing stable connectivity, high spectrum efficiency, and low latency. However, it relies significantly on the desires of both parties and makes compromises to provide data, necessitating a highly secure transmission network.

6. Open Challenges

First, as a fundamental requirement of 6G networks, *massive user density* of 10^7 active connections is expected to be served in a 1-square-kilometer area [205]. Supporting such an large number of concurrent connections is considered a challenging issue for MEC systems in upcoming applications because the limited resources at the MEC are expected to process a large amount of user traffic to meet high user demand. Moreover, each user request may ask for different service quality sets. To this end, multiple aspects of MEC systems must

be jointly optimized to simultaneously satisfy the system performance to improve user experience. In particular, an intelligent and powerful mechanism to efficiently exploit big data generated by dense user devices in real time is important for information-centric systems.

Second, the *data heterogeneity* of the spatiotemporal distribution of traffic originators makes it difficult to design optimal caching solutions to handle the entire user traffic. Spatiotemporal heterogeneity negatively impacts the convergence of optimal solutions, while user traffic properties continuously fluctuate in uncertain environments. For instance, pollution IoT devices in smart cities generate periodic report messages to the networks for processing, while transportation IoT devices monitor vehicular conditions and send their sensing data to the networks when predefined events occur, such as road jamming and accidents. This data traffic may be additionally affected by streaming flows from residents who use social video sharing services to disseminate their views. Obviously, each type of user traffic requests a specific caching approach for efficient processing.

Third, although *contextual adaptability* has been considered in recent studies, there is a lack of thorough investigation on practical scenarios with comprehensive constraints in realizing real implementations. In addition, next generation networks comprise numerous new user devices and applications. Hence, the contextual adaptability of MEC to information fusion should be addressed to achieve significant improvements in advance, as this feature is directly related to the service experience of the user. Moreover, intelligent contextual adaptability is vital to enable smart applications to personalize and localize services for individual circumstances.

Fourth, latency has been considered as one of the major metrics in caching techniques for real-time services. Consequently, latency continues to play an important role in future networks such as 6G and beyond, but with much higher requirements, that is, *ultralow latency*. Along with the rapid development of mobile cloudification and virtualization technologies [206], caching techniques empowered by these infrastructural capabilities are expected to provide information fusion in real time within ms delay. This stringent requirement becomes more challenging when incorporating other system performance and user experience optimization objectives, which the MEC must face and jointly handle.

Fifth, to improve user experience, modern content services aim to accommodate users with *high-fidelity* visualization and rich information. Obviously, high fidelity requires a large space to cache the content, powerful computational resources to process the data, and a wide wireless spectrum for user access. Such requirements become a big issue as relevant resources in networking devices are limited, while concurrent user connections exponentially increase. Hence, multi-resource management should be considered as one of the key targets for future research in MEC systems. Depending on particular application scenarios, the high fidelity can be traded with other metrics to achieve a balance between system performance and user experience. For instance, video resolution can be considered as the objective of a maximization problem, while latency and service availability act as strict constraints in video streaming systems assisted by edge caching technologies.

Sixth, *information autoprocessing* has recently become a dominant feature in intelligent environments that are equipped with AI capabilities. Under these circumstances, the information is refined at networking devices by extracting, adding, removing, and fusing, to generate the desired knowledge. To this end, AI has the advantage of automatically identifying data patterns and learning context. Although utilizing AI for this purpose has been proven feasible, further studies should be conducted to develop efficient and practical solutions. On the other hand, a tradeoff between autoprocessing accuracy and resource consumption costs in terms of time, space, and energy must be considered, especially when advanced AI algorithms are applied.

Last but not least, *security and privacy* remain open challenges in MEC studies as a result of the sensitivity of user data and internal/external attack threats rendering systems vulnerable. In particular, mutual authentication between user devices and MEC is important to initially protect their transactions against unauthorized entities. In addition, to efficiently exploit and process the information and context in user data, both system and user authorization and accounting must be stringently designed to manage user data privacy. To this end, AI and blockchain technologies are promising candidates for future research. Furthermore, security and privacy should be jointly considered while optimizing the system performance and user experience metrics.

7. Concluding Remarks

Edge caching has had a considerable influence on the growth of the telecommunication sector, notably in the development of new 5G/6G technologies and applications. The MEC approach enables data and information fusion to be processed on delivery at fusion nodes on edge servers. The growing development of data traffic and mobile devices, in addition to increasing user expectations, are the grounds for affirmation that MEC promises to prosper in the future. This paper was provided to give a comprehensive overview of state-of-the-art edge caching utilization in data processing and information fusion for delivery through mobile networks. The goal of this survey is to introduce the current status of MEC development by reviewing recent research and discoveries in many areas of MEC. We began by carefully investigating caching systems and their operations in order to demonstrate the function of MEC in information fusion with in-network computing capabilities. In addition, we have described the features that were used to evaluate the performance of three commonly used caching methods. By providing relevant works on optimizing MEC under various models and metrics, we help readers stay up to date on the most recent MEC development efforts. We discovered the rise of ML and AI technologies to boost the caching productivity of edge servers after reviewing these studies. However, despite their efficacy, the use of these algorithms might result in additional resource costs on edges. To this end, a discussion on difficulties is presented to address the crucial challenges in these areas. Although edge caching raises many concerns that need to

be solved, promising technologies such as AI and blockchain can be incorporated to build appropriate solutions for the challenges, and thus, these technologies deserve greater attention in future research.

Acknowledgement

The-Vinh Nguyen and Anh-Tien Tran contribute equally to this study. Nhu-Ngoc Dao and Sungrae Cho are the corresponding authors. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1G1A1008105).

References

[1] Cisco, Global–2021 Forecast Highlights, (Accessed on October 15, 2021) (2020).

[2] L. Bariah, L. Mohjazi, S. Muhaidat, P. C. Sofotasios, G. K. Kurt, H. Yanikomeroglu, O. A. Dobre, A prospective look: Key enabling technologies, applications and open research topics in 6G networks, IEEE Access 8 (2020) 174792–174820.

[3] Y. Ruan, C. Joe-Wong, On the economic value of mobile caching, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 984–993.

[4] S. Gu, Y. Wang, N. Wang, W. Wu, Intelligent optimization of availability and communication cost in satellite-UAV mobile edge caching system with fault-tolerant codes, IEEE Transactions on Cognitive Communications and Networking 6 (4) (2020) 1230–1241.

[5] F. Zhou, N. Wang, G. Luo, L. Fan, W. Chen, Edge caching in multi-UAV-enabled Radio Access Networks: 3D Modeling and Spectral efficiency optimization, IEEE Transactions on Signal and Information Processing over Networks 6 (2020) 329–341. doi:10.1109/TSIPN.2020.2986360.

[6] W. Yu, A. Najafi, Y. Nevarez, Y. Huang, A. Garcia-Ortiz, TAAC: Task allocation meets approximate computing for Internet of Things, in: 2020 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 2020, pp. 1–5.

[7] K. Cao, Y. Liu, G. Meng, Q. Sun, An overview on edge computing research, IEEE access 8 (2020) 85714–85728.

[8] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, D. O. Wu, Edge computing in industrial internet of things: Architecture, advances and challenges, IEEE Communications Surveys & Tutorials 22 (4) (2020) 2462–2488.

[9] B. Xia, C. Yang, T. Cao, Modeling and analysis for cache-enabled networks with dynamic traffic, arXiv preprint arXiv:1609.05586 (2016).

[10] Huawei, IBM, Intel, Nokia Networks, NTT Docomo, Vodafone, Mobile edge computing - introductory technical whitepaper, Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf, [Online; Accessed 15-Oct-2021] (2018).

[11] L. Chen, Y. Zhou, M. Jing, R. T. Ma, Thunder crystal: a novel crowdsourcing-based content distribution platform, in: Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, ACM, 2015, pp. 43–48.

[12] Y. Zhang, C. Jiang, B. Yue, J. Wan, M. Guizani, Information fusion for edge intelligence: A survey, Information Fusion 81 (2022) 171–186.

[13] U. Sa'ad, D. S. Lakew, S. Cho, Edge caching for content sharing in vehicular networks: Technical challenges, existing approaches, and future directions, in: 2021 International Conference on Information Networking (ICOIN), IEEE, 2021, pp. 770–775.

[14] Z. Xu, L. Zhou, S. C.-K. Chau, W. Liang, Q. Xia, P. Zhou, Collaborate or separate? Distributed service caching in mobile edge clouds, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 2066–2075.

[15] L. Li, G. Zhao, R. S. Blum, A survey of caching techniques in cellular networks: Research issues and challenges in content placement and delivery strategies, IEEE Communications Surveys & Tutorials 20 (3) (2018) 1710–1732.

[16] T. Hou, G. Feng, S. Qin, W. Jiang, Proactive content caching by exploiting transfer learning for mobile edge computing, International Journal of Communication Systems 31 (11) (2018) e3706.

[17] S. Bommaraveni, T. X. Vu, S. Chatzinotas, B. Ottersten, Active content popularity learning and caching optimization with hit ratio guarantees, IEEE Access 8 (2020) 151350–151359.

[18] A.-T. Tran, D. S. Lakew, T.-V. Nguyen, V.-D. Tuong, T. P. Truong, N.-N. Dao, S. Cho, Hit ratio and latency optimization for caching systems: A survey, in: 2021 International Conference on Information Networking (ICOIN), IEEE, 2021, pp. 577–581.

[19] M. Tauberg, Power law in popular media, Available: https://medium.com/@michaeltauberg/power-law-in-popular-media-7d7efef3fb7c, [Online; Accessed 30-Oct-2021] (Sep 2018).

[20] G. Paschos, E. Bastug, I. Land, G. Caire, M. Debbah, Wireless caching: Technical misconceptions and business barriers, IEEE Communications Magazine 54 (8) (2016) 16–22.

[21] C. Fricker, P. Robert, J. Roberts, N. Sbihi, Impact of traffic mix on caching performance in a content-centric network, in: 2012 Proceedings IEEE INFOCOM Workshops, IEEE, 2012, pp. 310–315.

[22] E. Baştuğ, M. Bennis, M. Debbah, Living on the edge: The role of proactive caching in 5G wireless networks, arXiv preprint arXiv:1405.5974 (2014).

[23] B. Bharath, K. G. Nagananda, H. V. Poor, A learning-based approach to caching in heterogenous small cell networks, IEEE Transactions on Communications 64 (4) (2016) 1674–1686.

[24] W.-X. Liu, J. Zhang, Z.-W. Liang, L.-X. Peng, J. Cai, Content popularity prediction and caching for ICN: A deep learning approach with SDN, IEEE access 6 (2018) 5075–5089.

[25] L. Xing, Z. Zhang, H. Lin, F. Gao, Content centric network with label aided user modeling and cellular partition, IEEE Access 5 (2017) 12576–12583.

[26] J. Xu, L. Chen, P. Zhou, Joint service caching and task offloading for mobile edge computing in dense networks, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 207–215.

[27] M. A. Maddah-Ali, U. Niesen, Decentralized coded caching attains order-optimal memory-rate tradeoff, IEEE/ACM Transactions on Networking (TON) 23 (4) (2015) 1029–1040.

[28] X. Wang, Z. Sheng, S. Yang, V. C. Leung, Tag-assisted social-aware opportunistic device-to-device sharing for traffic offloading in mobile social networks, IEEE Wireless Communications 23 (4) (2016) 60–67.

[29] N.-N. Dao, D. T. Ngo, N.-T. Dinh, T. V. Phan, N. D. Vo, S. Cho, T. Braun, Hit ratio and content quality tradeoff for adaptive bitrate streaming in edge caching systems, IEEE Systems Journal (2020).

[30] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, Web caching and Zipf-like distributions: Evidence and implications, in: INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 1, IEEE, 1999, pp. 126–134.

[31] J. Song, H. Song, W. Choi, Optimal caching placement of caching system with helpers, in: Communications (ICC), 2015 IEEE International Conference on, IEEE, 2015, pp. 1825–1830.

[32] Y. Guan, X. Zhang, Z. Guo, Prefcache: Edge cache admission with user preference learning for video content distribution, IEEE Transactions on Circuits and Systems for Video Technology 31 (4) (2020) 1618–1631.

[33] A. Tatar, M. D. De Amorim, S. Fdida, P. Antoniadis, A survey on predicting the popularity of web content, Journal of Internet Services and Applications 5 (1) (2014) 8.

[34] C. Wang, C. Liang, F. R. Yu, Q. Chen, L. Tang, Computation offloading and resource allocation in wireless cellular networks with mobile edge computing, IEEE Transactions on Wireless Communications 16 (8) (2017) 4924–4938.

[35] U. Niesen, M. A. Maddah-Ali, Coded caching with nonuniform demands, IEEE Transactions on Information Theory 63 (2) (2017) 1146–1158.

[36] M. Rim, C. G. Kang, Cache partitioning and caching strategies for device-to-device caching systems, IEEE Access 9 (2021) 8192–8211.

[37] H. Ahlehagh, S. Dey, Video-aware scheduling and caching in the radio

access network, IEEE/ACM Transactions on Networking (TON) 22 (5) (2014) 1444–1462.
[38] A. Brodersen, S. Scellato, M. Wattenhofer, Youtube around the world: geographic popularity of videos, in: Proceedings of the 21st international conference on World Wide Web, ACM, 2012, pp. 241–250.
[39] C. Zhong, M. C. Gursoy, S. Velipasalar, Deep reinforcement learning-based edge caching in wireless networks, IEEE Transactions on Cognitive Communications and Networking 6 (1) (2020) 48–61.
[40] Z. Qin, X. Gan, L. Fu, X. Di, J. Tian, X. Wang, Content delivery in cache-enabled wireless evolving social networks, IEEE Transactions on Wireless Communications 17 (10) (2018) 6749–6761.
[41] Z. Xu, S. Wang, S. Liu, H. Dai, Q. Xia, W. Liang, G. Wu, Learning for exception: Dynamic service caching in 5G-enabled MECs with bursty user demands, in: 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2020, pp. 1079–1089.
[42] S. Ju, Z. Zhang, X. Zhu, Video prediction strategy based on markov modified model, in: 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE), IEEE, 2020, pp. 105–109.
[43] S. Anokye, D. Ayepah-Mensah, A. M. Seid, G. O. Boateng, G. Sun, Deep reinforcement learning-based mobility-aware UAV content caching and placement in mobile edge networks, IEEE Systems Journal (2021).
[44] Y. Wang, C. Feng, T. Zhang, Y. Liu, A. Nallanathan, QoE based network deployment and caching placement for cache-enabling UAV networks, in: ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE, 2020, pp. 1–6.
[45] E. Baştuğ, M. Bennis, M. Kountouris, M. Debbah, Cache-enabled small cell networks: Modeling and tradeoffs, EURASIP Journal on Wireless Communications and Networking 2015 (1) (2015) 41.
[46] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, M. S. Hossain, Mobility-aware proactive edge caching for connected vehicles using federated learning, IEEE Transactions on Intelligent Transportation Systems (2020).
[47] A. Alioua, S. Simoud, S. Bourema, M. Khelifi, S.-M. Senouci, A stackelberg game approach for incentive V2V caching in software-defined 5G-enabled VANET, in: 2020 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2020, pp. 1–6.
[48] L. Zhao, H. Li, N. Lin, M. Lin, C. Fan, J. Shi, Intelligent content caching strategy in autonomous driving toward 6G, IEEE Transactions on Intelligent Transportation Systems (2021).
[49] B. Bai, L. Wang, Z. Han, W. Chen, T. Svensson, Caching based socially-aware D2D communications in wireless content delivery networks: a hypergraph framework, IEEE Wireless Communications 23 (4) (2016) 74–81.
[50] J. Ni, K. Zhang, A. V. Vasilakos, Security and privacy for mobile edge caching: challenges and solutions, IEEE Wireless Communications (2020).
[51] N. Abani, T. Braun, M. Gerla, Proactive caching with mobility prediction under uncertainty in information-centric networks, in: Proceedings of the 4th ACM Conference on Information-Centric Networking, ACM, 2017, pp. 88–97.
[52] T. X. Tran, D. V. Le, G. Yue, D. Pompili, Cooperative hierarchical caching and request scheduling in a cloud radio access network, IEEE Transactions on Mobile Computing (2018).
[53] J. Chen, H. Xing, X. Lin, S. Bi, Joint cache placement and bandwidth allocation for FDMA-based mobile edge computing systems, in: ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE, 2020, pp. 1–7.
[54] B. Nour, H. Khelifi, H. Moungla, R. Hussain, N. Guizani, A distributed cache placement scheme for large-scale information-centric networking, IEEE Network 34 (6) (2020) 126–132.
[55] S. O. Somuyiwa, D. Gündüz, A. Gyorgy, Reinforcement learning for proactive caching of contents with different demand probabilities, in: 2018 15th International Symposium on Wireless Communication Systems (ISWCS), IEEE, 2018, pp. 1–6.
[56] Z. Ding, P. Fan, G. K. Karagiannidis, R. Schober, H. V. Poor, NOMA assisted wireless caching: Strategies and performance analysis, IEEE Transactions on Communications (2018).
[57] M. Furqan, C. Zhang, W. Yan, A. Shahid, M. Wasim, Y. Huang, A collaborative hotspot caching design for 5G cellular network, IEEE Access 6 (2018) 38161–38170.
[58] H. Ben-Ammar, Y. Hadjadj-Aoul, G. Rubino, S. Ait-Chellouche, On the performance analysis of distributed caching systems using a customizable markov chain model, Journal of Network and Computer Applications (2019).
[59] W. Liu, Y. Jiang, S. Xu, G. Cao, W. Du, Y. Cheng, Mobility-aware video prefetch caching and replacement strategies in mobile-edge computing networks, in: 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), IEEE, 2018, pp. 687–694.
[60] J. Gao, S. Zhang, L. Zhao, X. Shen, The design of dynamic probabilistic caching with time-varying content popularity, IEEE Transactions on Mobile Computing 20 (4) (2020) 1672–1684.
[61] K. S. Kamath, N. Gupta, V. Shivaram, Comparison of caching policies in wireless networks, in: 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR), IEEE, 2018, pp. 1–3.
[62] A. Khosrozadeh, S. Pashmforoush, A. Akbari, M. Bagheri, N. Beikmahdavi, Presenting a novel page replacement algorithm based on LRU, Journal of Basic and Applied Scientific Research 2 (10) (2012) 10377–10383.
[63] G. Quan, K. Ji, J. Tan, LRU caching with dependent competing requests, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 459–467.
[64] J. S. Kushwah, J. K. Gupta, M. S. Yadav, H. Madan, Modified LRU algorithm to implement proxy server with caching policies, International Journal of Computer Science Issues (IJCSI) 8 (6) (2011) 352.
[65] J. Gu, W. Wang, A. Huang, H. Shan, Z. Zhang, Distributed cache replacement for caching-enable base stations in cellular networks, in: Communications (ICC), 2014 IEEE International Conference on, IEEE, 2014, pp. 2648–2653.
[66] Y. Chen, K. Shanmugam, A. G. Dimakis, From centralized to decentralized coded caching (2018). arXiv:1801.07734.
[67] W. Jiang, G. Feng, S. Qin, Optimal cooperative content caching and delivery policy for heterogeneous cellular networks, IEEE Transactions on Mobile Computing (1) (2017) 1–1.
[68] K. Guo, Z. Liang, R. Shi, C. Hu, Z. Li, Transparent learning: An incremental machine learning framework based on transparent computing, IEEE Network 32 (1) (2018) 146–151.
[69] M. Ji, G. Caire, A. F. Molisch, Fundamental limits of caching in wireless D2D networks, IEEE Transactions on Information Theory 62 (2) (2016) 849–869.
[70] Y. Fadlallah, A. M. Tulino, D. Barone, G. Vettigli, J. Llorca, J.-M. Gorce, Coding for caching in 5G networks, IEEE Communications Magazine 55 (2) (2017) 106–113.
[71] K. Shanmugam, M. Ji, A. M. Tulino, J. Llorca, A. G. Dimakis, Finite-length analysis of caching-aided coded multicasting, IEEE Transactions on Information Theory 62 (10) (2016) 5524–5537.
[72] M. A. Maddah-Ali, U. Niesen, Fundamental limits of caching, IEEE Transactions on Information Theory 60 (5) (2014) 2856–2867.
[73] M. Cheng, K. Wan, D. Liang, M. Zhang, G. Caire, A novel transformation approach of shared-link coded caching schemes for multiaccess networks, IEEE Transactions on Communications (2021).
[74] U. Niesen, M. A. Maddah-Ali, Coded caching for delay-sensitive content, in: Communications (ICC), 2015 IEEE International Conference on, IEEE, 2015, pp. 5559–5564.
[75] S. Gao, P. Dong, Z. Pan, G. Y. Li, Reinforcement learning based cooperative coded caching under dynamic popularities in ultra-dense networks, IEEE Transactions on Vehicular Technology 69 (5) (2020) 5442–5456.
[76] K. Wan, G. Caire, On coded caching with private demands, IEEE Transactions on Information Theory 67 (1) (2020) 358–372.
[77] S. Ahangary, H. Chitsaz, M. J. Sobouti, A. H. Mohajerzadeh, M. H. Yaghmaee, H. Ahmadi, Reactive caching of viral content in 5G networks, in: 2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet), IEEE, 2020, pp. 1–7.
[78] A. Sadeghi, F. Sheikholeslami, G. B. Giannakis, Optimal and scalable caching for 5G using reinforcement learning of space-time popularities, IEEE Journal of Selected Topics in Signal Processing 12 (1) (2018) 180–190.
[79] L. Ale, N. Zhang, H. Wu, D. Chen, T. Han, Online proactive caching in mobile edge computing using bidirectional deep recurrent neural network, IEEE Internet of Things Journal 6 (3) (2019) 5520–5530.
[80] A. Gharaibeh, A. Khreishah, B. Ji, M. Ayyash, A provably efficient on-

line collaborative caching algorithm for multicell-coordinated systems, IEEE Transactions on Mobile Computing 15 (8) (2016) 1863–1876.
[81] Y. Jiang, H. Feng, F.-C. Zheng, D. Niyato, X. You, Deep learning-based edge caching in fog radio access networks, IEEE Transactions on Wireless Communications 19 (12) (2020) 8442–8454.
[82] Y. Jiang, X. Chen, F.-C. Zheng, D. Niyato, X. You, Brain storm optimization-based edge caching in fog radio access networks, IEEE Transactions on Vehicular Technology 70 (2) (2021) 1807–1820.
[83] Y. Zhang, B. Feng, W. Quan, A. Tian, K. Sood, Y. Lin, H. Zhang, Cooperative edge caching: A multi-agent deep learning based approach, IEEE Access 8 (2020) 133212–133224.
[84] J. Chen, H. Wu, P. Yang, F. Lyu, X. Shen, Cooperative edge caching with location-based and popular contents for vehicular networks, IEEE Transactions on Vehicular Technology 69 (9) (2020) 10291–10305.
[85] C. Chen, J. Jiang, R. Fu, L. Chen, C. Li, S. Wan, An intelligent caching strategy considering time-space characteristics in vehicular named data networks, IEEE Transactions on Intelligent Transportation Systems (2021).
[86] Z. Xiaoqiang, Z. Min, W. Muqing, An in-network caching scheme based on betweenness and content popularity prediction in content-centric networking, in: Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 IEEE 27th Annual International Symposium on, IEEE, 2016, pp. 1–6.
[87] C. Zhong, M. C. Gursoy, S. Velipasalar, A deep reinforcement learning-based framework for content caching, in: Information Sciences and Systems (CISS), 2018 52nd Annual Conference on, IEEE, 2018, pp. 1–6.
[88] T. Ma, Y. Hao, W. Shen, Y. Tian, M. Al-Rodhaan, An improved web cache replacement algorithm based on weighting and cost, IEEE Access 6 (2018) 27010–27017.
[89] Y. Tang, K. Guo, J. Ma, Y. Shen, T. Chi, A smart caching mechanism for mobile multimedia in information centric networking with edge computing, Future Generation Computer Systems 91 (2019) 590–600.
[90] B. Banerjee, A. Kulkarni, A. Seetharam, Greedy caching: An optimized content placement strategy for information-centric networks, Computer Networks 140 (2018) 78–91.
[91] J. Roberts, N. Sbihi, Exploring the memory-bandwidth tradeoff in an information-centric network, in: Teletraffic Congress (ITC), 2013 25th International, IEEE, 2013, pp. 1–9.
[92] P. Pirozmand, G. Wu, B. Jedari, F. Xia, Human mobility in opportunistic networks: Characteristics, models and prediction methods, Journal of Network and Computer Applications 42 (2014) 45–58.
[93] L. Wu, S. Liu, H. Chen, N. Zhang, Using a novel grey system model to forecast natural gas consumption in China, Mathematical Problems in Engineering 2015 (2015).
[94] C. Bernardini, T. Silverston, F. Olivier, Mpc: Popularity-based caching strategy for content centric networks, in: IEEE International Conference onCommunications (ICC), 2013, IEEE, 2013, pp. 3619–3623.
[95] J. Hachem, N. Karamchandani, S. Diggavi, Content caching and delivery over heterogeneous wireless networks, in: Computer Communications (INFOCOM), 2015 IEEE Conference on, IEEE, 2015, pp. 756–764.
[96] N. di Pietro, E. C. Strinati, Proactive computation caching policies for 5G-and-beyond mobile edge cloud networks, in: 2018 26th European Signal Processing Conference (EUSIPCO), IEEE, 2018, pp. 792–796.
[97] M. Bilal, S.-G. Kang, A cache management scheme for efficient content eviction and replication in cache networks, IEEE Access 5 (2017) 1692–1701.
[98] X. Ma, A. Zhou, S. Zhang, S. Wang, Cooperative service caching and workload scheduling in mobile edge computing, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 2076–2085.
[99] A. O. Al-Abbasi, V. Aggarwal, TTLCache: taming latency in erasure-coded storage through TTL caching, IEEE Transactions on Network and Service Management 17 (3) (2020) 1582–1596.
[100] C. Fricker, P. Robert, J. Roberts, A versatile and accurate approximation for LRU cache performance, in: Teletraffic Congress (ITC 24), 2012 24th International, IEEE, 2012, pp. 1–8.
[101] H. Wu, J. Chen, W. Xu, N. Cheng, W. Shi, L. Wang, X. Shen, Delay-minimized edge caching in heterogeneous vehicular networks: A matching-based approach, IEEE Transactions on Wireless Communications 19 (10) (2020) 6409–6424.
[102] B. Zhou, Y. Cui, M. Tao, Stochastic content-centric multicast scheduling for cache-enabled heterogeneous cellular networks, IEEE Transactions on Wireless Communications 15 (9) (2016) 6284–6297.
[103] K. Poularakis, G. Iosifidis, V. Sourlas, L. Tassiulas, Exploiting caching and multicast for 5G wireless networks, IEEE Transactions on Wireless Communications 15 (4) (2016) 2995–3007.
[104] J. Erman, K. K. Ramakrishnan, Understanding the super-sized traffic of the super bowl, in: Proceedings of the 2013 conference on Internet measurement conference, ACM, 2013, pp. 353–360.
[105] D. Liu, C. Yang, Energy efficiency of downlink networks with caching at base stations, IEEE Journal on Selected Areas in Communications 34 (4) (2016) 907–922.
[106] N. Li, J.-F. Martinez-Ortega, V. H. Diaz, Distributed power control for interference-aware multi-user mobile edge computing: A game theory approach, IEEE Access 6 (2018) 36105–36114.
[107] Y. He, Z. Zhang, Y. Zhang, A big data deep reinforcement learning approach to next generation green wireless networks, in: GLOBECOM 2017-2017 IEEE Global Communications Conference, IEEE, 2017, pp. 1–6.
[108] H. Wu, J. Zhang, Z. Cai, F. Liu, Y. Li, A. Liu, Toward energy-aware caching for intelligent connected vehicles, IEEE Internet of Things Journal 7 (9) (2020) 8157–8166.
[109] L. Li, D. Shi, R. Hou, R. Chen, B. Lin, M. Pan, Energy-efficient proactive caching for adaptive video streaming via data-driven optimization, IEEE Internet of Things Journal 7 (6) (2020) 5549–5561.
[110] Y. Liu, Y. Wang, R. Sun, S. Meng, R. Su, Energy efficient downlink resource allocation for D2D-assisted cellular networks with mobile edge caching, IEEE Access 7 (2019) 2053–2067.
[111] Y. An, X. Luo, An in-network caching scheme based on energy efficiency for content-centric networks, IEEE Access 6 (2018) 20184–20194.
[112] Z. Zhang, Q. Zhang, J. Miao, F. R. Yu, F. Fu, J. Du, T. Wu, Energy-Efficient Secure Video Streaming in UAV-Enabled Wireless Networks: A Safe-DQN Approach, IEEE Transactions on Green Communications and Networking (2021) 1–14.
[113] C. Peng, S.-B. Lee, S. Lu, H. Luo, H. Li, Traffic-driven power saving in operational 3G cellular networks, in: Proceedings of the 17th annual international conference on Mobile computing and networking, ACM, 2011, pp. 121–132.
[114] L. Lei, L. You, G. Dai, T. X. Vu, D. Yuan, S. Chatzinotas, A deep learning approach for optimizing content delivering in cache-enabled HetNet, in: 2017 international symposium on wireless communication systems (ISWCS), IEEE, 2017, pp. 449–453.
[115] J. Liu, B. Bai, J. Zhang, K. B. Letaief, Content caching at the wireless network edge: A distributed algorithm via belief propagation, in: Communications (ICC), 2016 IEEE International Conference on, IEEE, 2016, pp. 1–6.
[116] C. Liang, F. R. Yu, Enhancing mobile edge caching with bandwidth provisioning in software-defined mobile networks, in: 2017 IEEE International Conference on Communications (ICC), IEEE, 2017, pp. 1–6.
[117] X. Li, X. Wang, S. Xiao, V. C. Leung, Delay performance analysis of cooperative cell caching in future mobile networks, in: Communications (ICC), 2015 IEEE International Conference on, IEEE, 2015, pp. 5652–5657.
[118] J. Li, Y. Chen, Z. Lin, W. Chen, B. Vucetic, L. Hanzo, Distributed caching for data dissemination in the downlink of heterogeneous networks, IEEE Transactions on Communications 63 (10) (2015) 3553–3568.
[119] F. Al-Turjman, Information-centric framework for the internet of things (IoT): Traffic modeling & optimization, Future Generation Computer Systems 80 (2018) 63–75.
[120] Z. Fan, Q. Wu, M. Zhang, R. Zheng, Popularity and gain based caching scheme for information-centric networks, International Journal of Advanced Computer Research 7 (30) (2017) 71.
[121] J. Dai, Z. Zhang, D. Liu, Proactive caching over cloud radio access network with user mobility and video segment popularity awared, IEEE Access 6 (2018) 44396–44405.
[122] R. Amer, M. M. Butt, M. Bennis, N. Marchetti, Inter-cluster cooperation for wireless D2D caching networks, IEEE Transactions on Wireless Communications 17 (9) (2018) 6108–6121.
[123] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng,

B. Hu, V. C. Leung, Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching, IEEE Internet of Things Journal (2018).
[124] L. C. Mutalemwa, S. Shin, A classification of the enabling techniques for low latency and reliable communications in 5G and beyond: AI-enabled edge caching, IEEE Access 8 (2020) 205502–205533.
[125] S. Zhang, J. Li, H. Luo, J. Gao, L. Zhao, X. S. Shen, Low-latency and fresh content provision in information-centric vehicular networks, IEEE Transactions on Mobile Computing (2020).
[126] Mark Jansen and Paula Beaton, 5G vs. 4G: How will the newest network improve on the last?, (Accessed on October 25, 2021) (2021).
[127] H. Huang, B. Liu, L. Chen, W. Xiang, M. Hu, Y. Tao, D2D-assisted VR video pre-caching strategy, IEEE Access 6 (2018) 61886–61895.
[128] C. C. Moallemi, B. Van Roy, Resource allocation via message passing, INFORMS journal on Computing 23 (2) (2011) 205–219.
[129] S. Boyd, L. Vandenberghe, Convex optimization, Cambridge university press, 2004.
[130] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, K. Ramakrishnan, Optimal content placement for a large-scale VoD system, IEEE/ACM Transactions on Networking 24 (4) (2016) 2114–2127.
[131] J. Zhang, X. Lin, X. Wang, Coded caching under arbitrary popularity distributions, IEEE Transactions on Information Theory 64 (1) (2018) 349–366.
[132] Y. He, S. Hu, Cache-enabled wireless networks with opportunistic interference alignment, arXiv preprint arXiv:1706.09024 (2017).
[133] M. Tao, E. Chen, H. Zhou, W. Yu, Content-centric sparse multicast beamforming for cache-enabled cloud RAN, IEEE Transactions on Wireless Communications 15 (9) (2016) 6118–6131.
[134] N. Golrezaei, P. Mansourifard, A. F. Molisch, A. G. Dimakis, Base-station assisted device-to-device communications for high-throughput wireless video networks, IEEE Transactions on Wireless Communications 13 (7) (2014) 3665–3676.
[135] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, S. N. Diggavi, Hierarchical coded caching, IEEE Transactions on Information Theory 62 (6) (2016) 3212–3229.
[136] M. Ji, G. Caire, A. F. Molisch, The throughput-outage tradeoff of wireless one-hop caching networks, IEEE Transactions on Information Theory 61 (12) (2015) 6833–6859.
[137] C. Zhan, G. Yao, Svc video delivery in cache-enabled wireless HetNet, IEEE Systems Journal (99) (2018) 1–4.
[138] X. Zhang, T. Lv, S. Yang, Near-optimal layer placement for scalable videos in cache-enabled small-cell networks, IEEE Transactions on Vehicular Technology 67 (9) (2018) 9047–9051.
[139] Y. Guo, Q. Yang, F. R. Yu, V. C. Leung, Cache-enabled adaptive video streaming over vehicular networks: A dynamic approach, IEEE Transactions on Vehicular Technology (2018).
[140] J. Leskovec, A. Krevl, {SNAP Datasets}:{Stanford} large network dataset collection (2015).
[141] J. Wu, B. Li, Keep cache replacement simple in peer-assisted vod systems, in: INFOCOM 2009, IEEE, IEEE, 2009, pp. 2591–2595.
[142] E. Baştuğ, M. Bennis, M. Debbah, A transfer learning approach for cache-enabled wireless networks, in: Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015 13th International Symposium on, IEEE, 2015, pp. 161–166.
[143] T. Lipp, S. Boyd, Variations and extension of the convex–concave procedure, Optimization and Engineering 17 (2) (2016) 263–287.
[144] D. Christopoulos, S. Chatzinotas, B. Ottersten, Cellular-broadcast service convergence through caching for coMP cloud RANs, in: Communications and Vehicular Technology in the Benelux (SCVT), 2015 IEEE Symposium on, IEEE, 2015, pp. 1–6.
[145] H. Schwarz, M. Wien, The scalable video coding extension of the H.264/AVC standard, IEEE Signal Processing Magazine 25 (2) (2008) 135.
[146] J. Qiao, Y. He, X. S. Shen, Proactive caching for mobile video streaming in millimeter wave 5G networks., IEEE Trans. Wireless Communications 15 (10) (2016) 7187–7198.
[147] M. Choi, J. Kim, J. Moon, Wireless video caching and dynamic streaming under differentiated quality requirements, IEEE Journal on Selected Areas in Communications 36 (6) (2018) 1245–1257.
[148] S. A. R. Zaidi, M. Ghogho, D. C. McLernon, Information centric modeling for two-tier cache enabled cellular networks, in: Communication Workshop (ICCW), 2015 IEEE International Conference on, IEEE, 2015, pp. 80–86.
[149] F. Lyu, J. Ren, N. Cheng, P. Yang, M. Li, Y. Zhang, X. Shen, LEAD: Large-scale edge cache deployment based on spatio-temporal WiFi traffic statistics, IEEE Transactions on Mobile Computing (2020).
[150] L. Sun, H. Pang, L. Gao, Joint sponsor scheduling in cellular and edge caching networks for mobile video delivery, IEEE Transactions on Multimedia (2018).
[151] M. Deghel, E. Bastug, M. Assaad, M. Debbah, On the benefits of edge caching for MIMO interference alignment, in: Signal Processing Advances in Wireless Communications (SPAWC), 2015 IEEE 16th International Workshop on, IEEE, 2015, pp. 655–659.
[152] M. Ji, G. Caire, A. F. Molisch, Optimal throughput-outage trade-off in wireless one-hop caching networks, in: 2013 IEEE International Symposium on Information Theory, IEEE, 2013, pp. 1461–1465.
[153] X.-Q. Pham, T.-D. Nguyen, V. Nguyen, E.-N. Huh, Joint service caching and task offloading in multi-access edge computing: A qoe-based utility optimization approach, IEEE Communications Letters 25 (3) (2020) 965–969.
[154] S. Bommaraveni, S. Gautam, T. X. Vu, S. Chatzinotas, On the exploration and exploitation trade-off in cooperative caching-enabled networks, in: 2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), IEEE, 2021, pp. 1–7.
[155] A. Asheralieva, D. Niyato, Combining contract theory and lyapunov optimization for content sharing with edge caching and device-to-device communications, IEEE/ACM Transactions on Networking 28 (3) (2020) 1213–1226.
[156] J. Feng, F. R. Yu, Q. Pei, J. Du, L. Zhu, Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems, IEEE Transactions on Wireless Communications 19 (6) (2020) 4321–4334.
[157] C.-Y. Wang, S. H. Lim, M. Gastpar, Information-theoretic caching: Sequential coding for computing, IEEE Transactions on Information Theory 62 (11) (2016) 6393–6406.
[158] S. B. Hassanpour, A. Diyanat, A. Khonsari, S. P. Shariatpanahi, A. Dadlani, Context-aware privacy preservation in network caching: An information theoretic approach, IEEE Communications Letters 25 (1) (2020) 54–58.
[159] Z. Hu, Z. Zheng, T. Wang, L. Song, X. Li, Game theoretic approaches for wireless proactive caching, IEEE Communications Magazine 54 (8) (2016) 37–43.
[160] X. Kang, R. Zhang, M. Motani, Price-based resource allocation for spectrum-sharing femtocell networks: A stackelberg game approach, IEEE Journal on Selected areas in Communications 30 (3) (2012) 538–549.
[161] Z. Su, Q. Xu, F. Hou, Q. Yang, Q. Qi, Edge caching for layered video contents in mobile social networks, IEEE Transactions on Multimedia 19 (10) (2017) 2210–2221.
[162] J. Li, W. Chen, M. Xiao, F. Shu, X. Liu, Efficient video pricing and caching in heterogeneous networks., IEEE Trans. Vehicular Technology 65 (10) (2016) 8744–8751.
[163] A. Roth, M. Sotomayor, Two-sided matching: A study in game-theoretic modeling and analysis, econometric society monograph series, cambridge university press, 1990 (1990).
[164] K. Hamidouche, W. Saad, M. Debbah, Many-to-many matching games for proactive social-caching in wireless small cell networks, in: Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2014 12th International Symposium on, IEEE, 2014, pp. 569–574.
[165] S. Müller, O. Atan, M. van der Schaar, A. Klein, Context-aware proactive content caching with service differentiation in wireless networks, IEEE Transactions on Wireless Communications 16 (2) (2017) 1024–1036.
[166] A. Gharaibeh, A. Khreishah, I. Khalil, An O (1)-competitive online caching algorithm for content centric networking, in: Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on, IEEE, 2016, pp. 1–9.
[167] S. Li, J. Xu, M. van der Schaar, W. Li, Trend-aware video caching through online learning, IEEE Transactions on Multimedia 18 (12) (2016) 2503–2516.
[168] E. Baştuğ, M. Bennis, E. Zeydan, M. A. Kader, I. A. Karatepe, A. S.

Er, M. Debbah, Big data meets telcos: A proactive caching perspective, Journal of Communications and Networks 17 (6) (2015) 549–557.
[169] Y. Zhang, K. Guo, J. Ren, Y. Zhou, J. Wang, J. Chen, Transparent computing: A promising network computing paradigm, Computing in Science & Engineering 19 (1) (2017) 7–20.
[170] Y. Wang, V. Friderikos, A survey of deep learning for data caching in edge network, in: Informatics, Vol. 7, Multidisciplinary Digital Publishing Institute, 2020, p. 43.
[171] Y. Zhang, Y. Li, R. Wang, J. Lu, X. Ma, M. Qiu, PSAc: Proactive sequence-aware content caching via deep learning at the network edge, IEEE Transactions on Network Science and Engineering 7 (4) (2020) 2145–2154.
[172] A. Ndikumana, N. H. Tran, K. T. Kim, C. S. Hong, et al., Deep learning based caching for self-driving cars in multi-access edge computing, IEEE Transactions on Intelligent Transportation Systems 22 (5) (2020) 2862–2877.
[173] M. Leconte, G. Paschos, L. Gkatzikis, M. Draief, S. Vassilaras, S. Chouvardas, Placing dynamic content in caches with small population, in: INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE, IEEE, 2016, pp. 1–9.
[174] E. Bastug, M. Bennis, M. Debbah, Anticipatory caching in small cell networks: A transfer learning approach, in: 1st KuVS Workshop on Anticipatory Networks, 2014.
[175] Y. Wei, Z. Zhang, F. R. Yu, Z. Han, Joint user scheduling and content caching strategy for mobile edge networks using deep reinforcement learning, in: 2018 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2018, pp. 1–6.
[176] I. Grondman, M. Vaandrager, L. Busoniu, R. Babuska, E. Schuitema, Efficient model learning methods for actor–critic control, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (3) (2012) 591–602.
[177] Q. Xu, Z. Su, R. Lu, Game theory and reinforcement learning based secure edge caching in mobile social networks, IEEE Transactions on Information Forensics and Security 15 (2020) 3415–3429.
[178] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529.
[179] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of Go with deep neural networks and tree search, nature 529 (7587) (2016) 484.
[180] X. Wang, C. Wang, X. Li, V. C. Leung, T. Taleb, Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching, IEEE Internet of Things Journal 7 (10) (2020) 9441–9455.
[181] Y. Dai, D. Xu, K. Zhang, S. Maharjan, Y. Zhang, Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks, IEEE Transactions on Vehicular Technology 69 (4) (2020) 4312–4324.
[182] C. Chen, Y. Zhang, Z. Wang, S. Wan, Q. Pei, Distributed computation offloading method based on deep reinforcement learning in icv, Applied Soft Computing 103 (2021) 107108.
[183] G. Kossinets, D. J. Watts, Empirical analysis of an evolving social network, Science 311 (5757) (2006) 88–90. arXiv:https://science.sciencemag.org/content/311/5757/88.full.pdf, doi:10.1126/science.1116869.
URL https://science.sciencemag.org/content/311/5757/88
[184] A.-T. Tran, N.-N. Dao, S. Cho, Bitrate adaptation for video streaming services in edge caching systems, IEEE Access 8 (2020) 135844–135852.
[185] H. Aftab, J. Shuja, W. Alasmary, E. Alanazi, Hybrid DBSCAN based community detection for edge caching in social media applications, in: 2021 International Wireless Communications and Mobile Computing (IWCMC), IEEE, 2021, pp. 2038–2043.
[186] X. He, K. Wang, H. Lu, W. Xu, S. Guo, Edge qoe: Intelligent big data caching via deep reinforcement learning, IEEE Network 34 (4) (2020) 8–13.
[187] D. Singh, S. Kumar, S. Kapoor, An explore view of web caching techniques (2011).
[188] MaxCDN Support Site, What is proxy caching?, Available: https://www.maxcdn.com/one/visual-glossary/proxy-caching/, [Online; Accessed 15-Oct-2021] (Sep 2018).
[189] G. Hasslinger, K. Ntougias, F. Hasslinger, O. Hohlfeld, Fast and efficient web caching methods regarding the size and performance measures per data object, in: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), IEEE, 2019, pp. 1–7.
[190] G. Hasslinger, K. Ntougias, F. Hasslinger, O. Hohlfeld, General knapsack bounds of web caching performance regarding the properties of each cacheable object, in: 2020 IFIP Networking Conference (Networking), IEEE, 2020, pp. 821–826.
[191] S. Zhang, J. Liu, Optimal probabilistic caching in heterogeneous IoT networks, IEEE Internet of Things Journal 7 (4) (2020) 3404–3414.
[192] S. Sheng, P. Chen, Z. Chen, L. Wu, H. Jiang, Edge caching for IoT transient data using deep reinforcement learning, in: IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2020, pp. 4477–4482.
[193] H. Wang, Y. Li, X. Zhao, F. Yang, An algorithm based on markov chain to improve edge cache hit ratio for blockchain-enabled IoT, China Communications 17 (9) (2020) 66–76.
[194] Ericsson, Video streaming to the extreme, Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/articles/streaming-video, (Accessed on November 1, 2021) (2018).
[195] R. Sun, Y. Wang, N. Cheng, H. Zhou, X. Shen, QoE driven BS clustering and multicast beamforming in cache-enabled C-RANs, in: 2018 IEEE International Conference on Communications (ICC), IEEE, 2018, pp. 1–6.
[196] B. Jedari, G. Premsankar, G. Illahi, M. Di Francesco, A. Mehrabi, A. Ylä-Jääski, Video caching, analytics, and delivery at the wireless edge: A survey and future directions, IEEE Communications Surveys & Tutorials 23 (1) (2020) 431–471.
[197] F. Wang, F. Wang, J. Liu, R. Shea, L. Sun, Intelligent video caching at network edge: A multi-agent deep reinforcement learning approach, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 2499–2508.
[198] A. Zhang, Q. Li, Y. Chen, X. Ma, L. Zou, Y. Jiang, Z. Xu, G.-M. Muntean, Video super-resolution and caching–an edge-assisted adaptive video streaming solution, IEEE Transactions on Broadcasting (2021).
[199] P. Maniotis, N. Thomos, Tile-based edge caching for 360° live video streaming, IEEE Transactions on Circuits and Systems for Video Technology (2021).
[200] Y. Liu, Q. Chang, M. Peng, T. Dang, W. Xiong, Virtual reality streaming in blockchain enabled fog radio access networks, IEEE Internet of Things Journal (2021).
[201] R. Zhang, F. R. Yu, J. Liu, T. Huang, Y. Liu, Deep reinforcement learning (DRL)-based device-to-device (D2D) caching with blockchain and mobile edge computing, IEEE Transactions on Wireless Communications 19 (10) (2020) 6469–6485.
[202] S. Mumtaz, K. M. S. Huq, J. Rodriguez, Direct mobile-to-mobile communication: Paradigm for 5G, IEEE Wireless Communications 21 (5) (2014) 14–23.
[203] R. Li, Y. Zhao, C. Wang, X. Wang, V. C. Leung, X. Li, T. Taleb, Edge caching replacement optimization for D2D wireless networks via weighted distributed dqn, in: 2020 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2020, pp. 1–6.
[204] S. S. Kafıloğlu, G. Gür, F. Alagöz, Cooperative caching and video characteristics in D2D edge networks, IEEE Communications Letters 24 (11) (2020) 2647–2651.
[205] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland, F. Tufvesson, 6G wireless systems: Vision, requirements, challenges, insights, and opportunities, Proceedings of the IEEE 109 (7) (2021) 1166–1199.
[206] N.-N. Dao, W. Na, S. Cho, Mobile cloudization storytelling: Current issues from an optimization perspective, IEEE Internet Computing 24 (1) (2020) 39–47.