TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 14/20 pp818–842
DOI: 10.26599/TST.2023.9010051
Volume 29, Number 3, June 2024

A Survey of Edge Caching: Key Issues and Challenges

Hanwen Li, Mingtao Sun, Fan Xia, Xiaolong Xu*, and Muhammad Bilal

Abstract: With the rapid development of mobile communication technology and intelligent applications, the quantity of mobile devices and data traffic in networks have been growing exponentially, which poses a great burden to networks and brings huge challenge to servicing user demand. Edge caching, which utilizes the storage and computation resources of the edge to bring resources closer to end users, is a promising way to relieve network burden and enhance user experience. In this paper, we aim to survey the edge caching techniques from a comprehensive and systematic perspective. We first present an overview of edge caching, summarizing the three key issues regarding edge caching, i.e., where, what, and how to cache, and then introducing several significant caching metrics. We then carry out a detailed and in-depth elaboration on these three issues, which correspond to caching locations, caching objects, and caching strategies, respectively. In particular, we innovate on the issue "what to cache", interpreting it as the classification of the "caching objects", which can be further classified into content cache, data cache, and service cache. Finally, we discuss several open issues and challenges of edge caching to inspire future investigations in this research area.

Key words: edge caching; edge computing; caching location; caching object; caching strategy; 5G network architecture; Internet of Things (IoT)

1 Introduction

In recent years, with the rapid development of mobile communication technology and intelligent devices, the quantity of mobile devices and data traffic have been growing explosively. According to the latest report of Cisco[1], by 2023, there will be 5.3 billion Internet users in total, accounting for 66% of the world population. Meanwhile, the number of IP-connected devices is expected to reach 29.3 billion, which is more than three times the world population.

The arrival of the 5G era not only brings better experience to users, but also injects a flow of fresh vitality into the implementation of the Internet of Everything (IoE)[2, 3]. According to Metcalfe's Law[4, 5], as more things, people, and data become connected to the Internet, the power of the Internet (i.e., the network of networks) is growing exponentially. In the IoE era,

• Hanwen Li is with School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China. E-mail: hanwenli713@gmail.com.
• Mingtao Sun is with Shandong Provincial University Laboratory for Protected Horticulture, Weifang University of Science and Technology, Weifang 262700, China. E-mail: sunmingtao@wfust.edu.cn.
• Fan Xia is with Reading Academy, Nanjing University of Information Science and Technology, Nanjing 210044, China. E-mail: xiafan307@gmail.com.
• Xiaolong Xu is with School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China. E-mail: xlxu@ieee.org.
• Muhammad Bilal is with Department of Computer Engineering, Hankuk University of Foreign Studies, Yongin-si 17035, Republic of Korea. E-mail: mbilal@hufs.ac.kr.
* To whom correspondence should be addressed.
Manuscript received: 2023-02-25; revised: 2023-05-07; accepted: 2023-05-23

Internet of Vehicles (IoV), intelligent health care, Augmented Reality (AR), Virtual Reality (VR), and many other new Internet of Things (IoT) applications are generating, while simultaneously requesting and consuming data and traffic every minute. The demand of mobile users for data transmission rate and service quality is also growing. Despite the rising computing capacity of smart mobile devices, they may still be incapable of processing computationally intensive applications, such as VR, AR, facial recognition, etc. in a short amount of time. Additionally, the battery consumption associated with operating apps involving high computational power needs remains a key barrier which prevents mobile consumers from fully enjoying these services[6].

Under this background, Edge Computing (EC), as a decentralized computing architecture, transfers the computation tasks from the center of the network to edge nodes closer to users, thus providing users with computing services with lower latency and higher security[7]. Edge caching, which integrates traditional caching methods and mechanisms into the EC infrastructure, refers to the practice of employing intermediary storage between traditional large-scale data centers and ultimate users who access resources. By moving memory storage nearer to end users, edge caching relieves stress on the network and improves content delivery performance, which is an essential research topic in EC.

Cache, in its original sense, refers to a type of high-speed memory which exists in CPU and can be accessed faster than ordinary Random Access Memory (RAM). The data and instructions in memory that are most frequently accessed by the CPU are duplicated into the cache, so that the speed difference between the CPU and the memory can be bridged. The cache is one of the most important factors for all modern computer systems to achieve high performance. Nowadays, the meaning of "cache" has transformed from one of necessary computer components to describing the idea and process of storing data for later access[8], which has become an indispensable technique for storing temporary data or files for speedy search by users for a long term. Meanwhile, the cache idea has been expanded to many other fields, including network and EC[9]. Edge caching can be regarded as the combination of the caching idea and mechanism with the technology and architecture of EC, as shown in Fig. 1, which is not only one of the key technologies in EC, but also an important application of EC technique per se.

As presented in Ref. [10], in fact, despite the variety of demands for different types of contents in the network, only a tiny percentage of them are frequently required. In other words, the great majority of requested contents are redundant, and the repeated transmission of these popular ones will cause a significant amount of duplicate traffic loads, thus leading to low transmission efficiency and excessive energy consumption. Therefore, by caching popular contents on edge devices, users' requests for the same content can be accommodated easily without the need for superfluous transmissions from remote servers[11]. By putting the resource closer to end users, edge caching can reduce transmission and calculation latency, improving users' Quality of Experience (QoE) dramatically.

So why does edge caching have such capability? To illustrate, consider two different scenarios of the international newspaper publication, as shown in Figs. 2a and 2b. In Scenario 1, all the printing is done in one place, which is called the headquarters. Then, the printed papers are sent all over the world. However, in Scenario 2, the headquarters first sends a master copy to presses around the world. Then the presses worldwide will make copies of the papers based on the master copy and deliver them locally. Considering the cost and efficiency, the latter method is more sensible than the former one. In former scenario, since all the publishing work is accomplished in just one place, the shipping costs per subscriber will offset profits greatly,

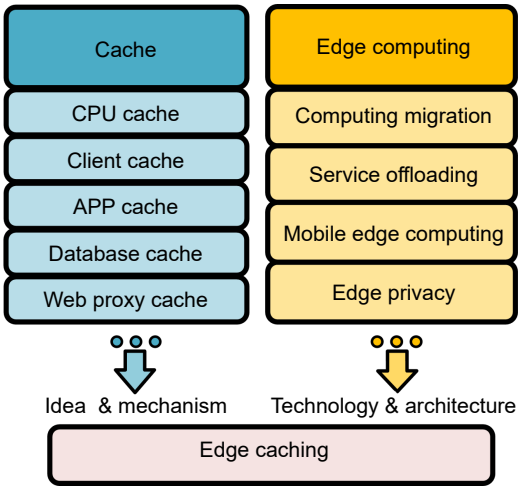

Fig. 1 Relationship between cache, EC, and edge caching.

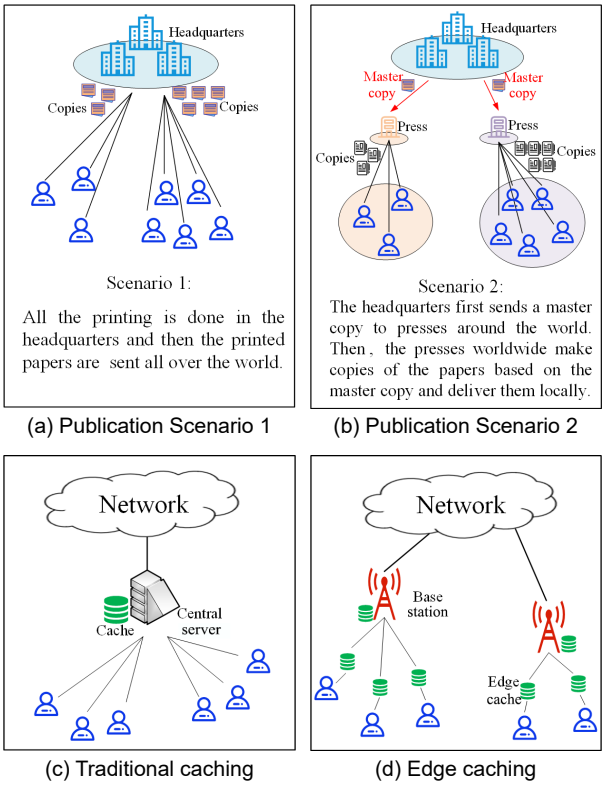

Fig. 2 Comparison between publication Scenario 1 and publication Scenario 2, and traditional caching and edge caching.

and the delays in receiving paper may cause customers to cancel their subscriptions. Therefore, in practice, most wise publishers will choose the latter method, because in this way, those local presses can print many newspapers every day and deliver them timely, thus maximizing the use of their facilities to improve profits. The caching can be thought of as a business for publishing international newspapers. As to edge caching, edge devices can be seen as those local presses, which deliver newspapers (e.g., web content) from popular publishers (content producers) to many end users. The simple comparison between traditional caching and edge caching is shown in Figs. 2c and 2d. In fact, edge caching has played a critical role in assisting modern Content Delivery Networks (CDN) and telecoms carriers, who provide web services to billions of users[12].

As to related surveys on edge caching, we have made a collection of them, and the summary and comparison of these works are concluded in Table 1. As shown in Table 1, although many works have attempted to survey various issues in edge caching, they have been limited to specific domains. From this point of view, a more comprehensive review of edge caching is necessary, in order to elaborate this topic in a more general sense, which is what this paper attempts to achieve. Moreover, regarding the definition of edge caching and the summary and specific content of key issues, different surveys have given different statements, without reaching consensus on many issues.

In this case, the purpose of this survey is to provide a clear picture of edge caching from a comprehensive and systematic perspective, with the hope that it will be a useful guide to those wishing to pursue research within this exciting field. To that end, in this section, the definition of edge caching in general sense has been given, while its advantages have been illustrated, and the rest of this paper will be organized as follows. Section 2 provides an overview of edge caching, summarizing the three key issues regarding edge caching and introducing some important metrics. Section 3 introduces the possible edge caching locations where cache storage can be located, along with existing researches concerned. In Section 4, edge caching objects are classified into three categories and elaborated respectively. The comparison of two pairs of edge caching strategies is discussed in Section 5. Section 6 discusses several challenges and open issues of edge caching. Section 7 concludes the survey.

2 Edge Caching: The Overview

In comparison to traditional centralized web caching, edge caching techniques are more advanced, yet complex. When conducting research into this area, it is important to identify its key questions first. Therefore, in this section, we will discuss and conclude the key issues regarding edge caching at first. Furthermore, when designing edge caching and considering its application into specific scenarios, the performance metrics are the primary reference, which will be mentioned repeatedly throughout the rest of this paper. Hence, after the key issues, we will attempt to define and introduce those most vital and commonly used performance metrics of edge caching from various aspects briefly.

2.1 Key issues regarding edge caching

As to relative surveys mentioned in Table 1, we find that almost every one of them has discussed edge caching in detail from the following three aspects, or some of them, as shown in Table 2. Therefore, in this

Table 1 Summary and comparison of relative surveys.

Reference	Year	Domain	Issue
[13]	2016	Wireless networks	Content popularity and user preference; content placement and delivery; key differences between wired and wireless caching; differences among different caching locations
[14]	2018	Cellular networks	Caching techniques in different cellular networks; caching algorithms from three aspects: content placement, content delivery, and joint placement and delivery; performance metrics
[15]	2019	Mobile edge caching	Caching locations; caching criteria; caching schemes; caching process
[16]	2019	Radio Access Networks (RAN) for IoT	Deployment location; content placement strategy; coded caching; hierarchical edge cache structure
[17]	2020	Mobile edge caching	Caching locations; cache replacement strategies; caching system behavior/performance; wireless networks caching optimization
[18]	2020	Edge intelligence	Preliminary of caching; cache deployment; cache replacement
[19]	2021	Mobile edge caching	Total process of edge caching: caching placement optimization; caching policy design; caching content delivery process
[20]	2021	5G and beyond 5G edge networks	Application of Machine Learning (ML) techniques for edge caching; caching strategy (policy, location, and replacement); edge network (type and delivery strategy)
[21]	2022	Edge caching in IoT smart environments	In-network caching solutions based on Named Data Networking (NDN) paradigm; possible interplay of NDN-based edge caching policies with Software-Defined Networking (SDN)
[22]	2023	Mobile edge caching	Social-aware edge caching mechanisms; users' social and behavioral characteristics effective in caching strategies; taxonomy of social-aware edge caching approaches

paper, we will follow this main classification pattern, where the key issues regarding edge caching technology can be mainly summarized into three aspects: (1) where to cache; (2) what to cache; and (3) how to cache, which correspond to the caching locations, caching objects, and caching strategies, respectively, as shown in Fig. 3.

2.1.1 Where to cache

Where to cache refers to the selection of caching locations, namely the places where caches are deployed or where caching behavior occurs. Researches on this issue have been very sufficient, and it is generally accepted that the edge caching locations can be classified into two broad categories[15, 16, 18]: (a) caching at Base Stations (BSs); (b) caching at end devices. Caching at various locations have distinct characteristics, and we will provide a detailed discussion about this in Section 3.

2.1.2 What to cache

From existing studies, different authors have different interpretations of "what to cache" is. Some think of it as the selection strategy for caching content, namely cache content placement strategy[16], while others interpret it as the classification of the types of caching content[15, 18]. In this paper, we prefer the latter interpretation. Nevertheless, we will not refer to "what

Table 2 Key issues covered by relative surveys.

Reference	Where to cache	What to cache	How to cache
[13]	✓		✓
[14]	✓		✓
[15]	✓	✓	✓
[16]	✓	✓	✓
[17]	✓		✓
[18]	✓	✓	✓
[19]		✓	✓
[20]	✓		✓
[21]		✓	✓
[22]		✓	✓

to cache" as the classification of the types of "caching content", but rather "caching objects". A novel classification pattern is proposed, and according to this pattern, the content cache is merely seen as one category of the classification. More specifically, the caching objects can be divided into three categories: (1) content cache; (2) data cache; and (3) service cache, about which we will provide a detailed elaboration in Section 4.

2.1.3 How to cache

How to cache refers to the design of caching strategies. From different perspectives, the edge caching strategies can be summarized into different categories. Conventional web caching usually adopts a reactive and centralized caching strategy, which is simpler but not enough to cope with the complex network environment where edge caching system is located. To deal with this problem, many existing researches have attempted to innovate on the traditional caching strategies, or to propose novel caching strategies, namely proactive caching and distributed caching. In this paper, the edge caching strategies issue will be formulated from two different perspectives: (1) reactive caching vs. proactive caching; (2) centralized caching vs. distributed caching, which will be introduced in Section 5.

2.2 Performance metrics

Due to the integration of caching mechanism into EC infrastructure, and the fact that edge caching is mainly deployed in network environment, performance metrics of edge caching can be categorized into: (1) cached-based metrics and (2) system metrics, from the perspective of the cache itself and the system where the edge caching resides severally. The summary of these significant performance metrics is shown in Table 3.

2.2.1 Cached-based metrics

Cache-based metrics are standard metrics used to measure the efficiency and the performance of caching techniques by measuring whether a caching strategy is able to store and hold the required content. Cache-based metrics are typically calculated on a node basis.

(1) **Cache hit ratio:** A cache hit occurs when an end user submits a request to the network while the required content exists in the cache exactly. Then, the desired content will be sent from the cache that hits directly to the user. CHR is defined as the percentage of requests that can be served by the cache, which reflects the load reduction of a server due to caching[93]. On the other hand, if the cache does not hit, namely the desired content does not exist in the cache, we call this situation a cache miss. In this case, the desired content will be transmitted to the user from the server, and generally the corresponding data will be loaded into the cache for future access.

Since the edge cache is closer to the user in comparison with servers, the overhead of transferring content from the former to end users is much less than

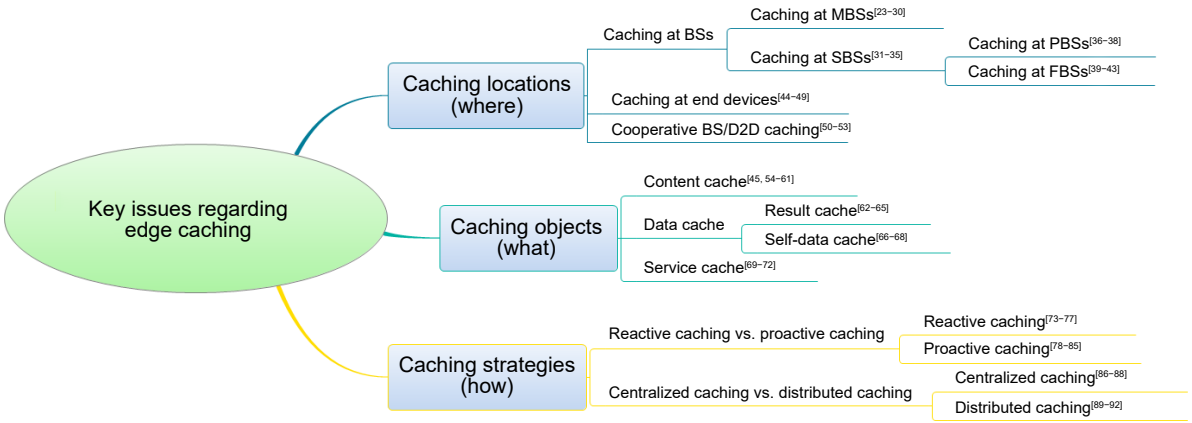

Fig. 3 Key issues regarding edge caching.

Table 3 Summary of significant performance metrics in edge caching.

Category	Metric	Definition
Cached-based metrics	Cache Hit Ratio (CHR)	Percentage of requests that can be satisfied by the cache.
	Cache Replacements Rate (CRR)	Ratio of the size of replaced content to the total cache capacity over a period of time.
	Popularity	Ratio of the times a certain item has been fed back by different users to the total number of users.
	Request probability	Probability that a certain item is requested.
System metrics	Network delay	Time duration experienced by consumers between the time when the file is requested until delivery.
	System throughput	Amount of data that a system successfully transmits per unit time.
	Backhaul cost	Cost incurred by introducing the backhaul link, which refers to the communication link connecting the BSs to each other or to the core network aimed at saving transmission time or reducing costs.
	Energy Efficiency (EE)	Ratio of the effective information transmission rate to the signal transmit power. (For the communication) Ratio of the total network throughput to the overall energy consumption. (For the network)
	Spectral Efficiency (SE)	Ratio of the effective information transmission rate to the channel bandwidth.

that of the latter. Therefore, preferable edge caching policies should correlate to an increase in the metric CHR, because it represents that more demands from users can be handled by the edge cache without duplicate transmissions from remote servers[94], so that the transmission efficiency and users' QoE can be improved.

(2) **Cache replacements rate:** Since the storage resource of edge devices is limited, when the cache capacity is full, the caching replacement algorithm must decide which items to discard to make room for the new ones. CRR is defined as the ratio of the size of replaced content to the total cache capacity over a period of time.

The more frequently content is replaced in the cache, the shorter the cache lifespan of popular content on nodes, resulting in a lower CHR and higher latency. Under the condition of fixed resources, more efficient caching replacement strategy with lower CRR is a necessity, which will track more usage information so that the contents of the cache can be fully exploited to improve the CHR for a given cache size.

(3) **Popularity and request probability:** When designing a caching strategy, the popularity of content needs to be considered. If contents with higher popularity are cached, they will be hit with a higher probability, thus increasing CHR and decreasing CRR. There are various ways to measure the popularity, and usually a ratio or a certain probability is calculated. In this way, popularity of an item is defined as the ratio of the times it has been fed back by different users to the total number of users, which can also be referred to click-through rate, viewing rate and completion rate depending on the form of feedback.

Having quantified the item popularity, it can be taken into account as to caching strategy design. In general, just a few popular files will be requested frequently by a large number of users at various periods on the network. In Ref. [95], it is confirmed that the distribution of web requests generally follows a Zipf-like distribution, where the relative probability of a request for the i-th most popular page is proportional to $1/i^{\alpha}$, with α denoting the Zipf exponent that describes the likelihood of content repetition. Many aspects of the Internet are governed by Zipf's law, and observations of Zipf distributions have significant meaning for the design and function of the Internet[96]. In many other studies, like Ref. [97], it is verified that the popularity distribution of files, such as videos, also follows the same Zipf-like distribution.

2.2.2 System metrics

Since edge caching is often applied in systems like network environments, when designing a system with edge caching, in addition to measuring the cache-based metrics, it is necessary to consider the performance metrics of the whole system, namely system metrics.

(1) **Network delay:** Network delay, also referred to as Round-Trip Time (RTT) or latency, represents the time duration experienced by users between the time when the request is sent until delivery. Specifically, the

network delay can generally be divided into four parts: (1) processing delay: the time it takes for routers to process data; (2) queuing delay: the time spent by data in routing queues; (3) transmission delay: the time required for a data block to enter the transmission medium from the node when sending data; (4) propagation delay: the time required for a signal to its destination. In this way, the network delay can be defined as the sum of these four types of delay.

Network delay has a significant impact on the user experience and is crucial for delay-sensitive services (e.g., interactive entertainment, online education, and media creation). For this reason, delay and traffic reduction comprise the primary objectives of in-network caching.

(2) **System throughput:** System throughput is defined as the amount of data that a system successfully transmits per unit time. In other words, it represents the actual transmission rate of the system. Throughput is one of important metrics to analyze and measure the network performance, which is critical in the design of a system with edge caching. To take video files, the most requested content in cellular networks as an example, users' QoE in video service mainly depends on throughput particularly.

When addressing equity across various users, an important metric utilized in designing caching strategies is the minimum average user throughput under the constraint of a network fault probability, which implies the possibility that a user demand cannot be fulfilled[98].

(3) **Backhaul cost:** In wireless communication, backhaul refers to the use of long-distance routing to transfer data from the network, the purpose of which is to save transmission time or reduce costs. For network architecture, backhaul is an important part of the network, referring to the links that connects BSs to each other or to the center network, usually made of pricey optical fiber, responsible for gathering data traffic from BSs and transmitting it to the metro/aggregation network[99].

Future wireless services with high data rate require extremely high backhaul bandwidth, which will result in high backhaul costs, thereby limiting the benefits of small cell installations otherwise. Therefore, it is critical that systems be designed with cost-efficient backhaul architecture. In an effort to alleviate the backhaul cost and capacity bottleneck, demand for lifting backhaul cost efficiency is becoming as vital as investment in radio infrastructure[100].

(4) **Energy efficiency:** In 5G wireless networks, with the addition of millions of BSs and billions of connected devices, the need for energy-efficient system design and operation will be even more compelling[101, 102]. Therefore, when designing a system with edge caching, it is important to take the system power consumption into consideration. In general, the power consumption from BSs and user devices is dominant. The former determines the energy cost for the operators, while the latter affects the battery life of user devices, which will affects users' QoE indirectly.

Nevertheless, in order to measure the energy metric of a system, besides power consumption, which shows the absolute value of the energy consumed and with the unit of Joule, another more effective measure is the EE. EE describes the number of transmission bits that can be obtained when the system consumes unit energy, representing system's utilization efficiency of energy resources. In the communication field, EE is defined as the ratio of the effective information transmission rate to the signal transmit power, with the unit of bit/J. For the network, EE can also be defined as the ratio of the total network throughput to the overall energy consumption[103].

(5) **Spectral efficiency:** In addition to EE, another important metric to measure the efficiency of a system is SE. SE is used to measure the effectiveness of the system, and describes how much capacity can be provided, representing the system's utilization efficiency of spectrum resources. It is defined as the ratio of the effective information transmission rate to the channel bandwidth, which can be viewed as the bits per second per hertz ($\text{bit}\cdot\text{s}^{-1}\cdot\text{Hz}^{-1}$) supported by the system[104].

Both EE and SE are the most commonly used and critical metrics for evaluating system efficiency, thus deserving consideration when designing a system with edge caching. Many works, like Refs. [105, 106], have also explored the tradeoffs between EE and SE, in order to optimize the system performance.

3 Caching Location

As demonstrated in Ref. [107], caching techniques which store content in caching devices at the edge of network for future usage are strong candidates for reducing backhaul traffic. A significant portion of the backhaul traffic stems from the duplicate transmission of popular content to multiple users[108]. Thanks to

edge caching, the redundant traffic could be reduced significantly. A simplified architecture of edge caching network system is shown in Fig. 4, where the edge caching is mainly deployed in BSs and end devices. Caching at different locations have different characteristics, which will be discussed in the following subsections.

3.1 Caching at BSs

According to 3GPP[109, 110], based on power and capacity, wireless BSs can be divided into two main categories, namely Macro BS (MBS) and Small BS (SBS). Further, the SBS can be subdivided into Pico BS (PBS) and Femto BS (FBS). The configurations of different BSs are compared in Table 4. As shown in Table 4, different types of BSs differ greatly in their characteristics, and therefore play different roles in edge caching. The deployment of cache in BSs abstracts the BS network into a distributed model, which can considerably relieve the burden on the core link. Distributive storage at the caching-enabled BSs has the potential to reduce the traffic load in future cellular networks[23].

Heterogeneous Network (HetNet), which coordinates the deployment of different BSs so that they can work synergistically, has been introduced to provide high data throughput, strong mobility, and high-quality user experience[24]. HetNet has been deemed as a promising architectural technique for 5G, where SBSs that address high-density access demands in small areas (e.g., PBSs and FBSs), are deployed intensively and work cooperatively with MBSs, which fulfill access requirements in wide area. In a cache-enabled HetNet, caching can occur at BSs of different types in order to provide users with service with more convenience and higher quality. Edge caching at different types of BSs will be discussed in the following subsections.

3.1.1 Caching at MBSs

Since the amount of MBS in the HetNet is usually limited because of its high cost and outdoor application scenarios, with respect to caching at MBSs, most current works have focused on the framework of caching at HetNet where a small number of MBSs work in conjunction with densely deployed SBSs.

Given the limited storage capacity, in order to achieve higher caching efficiency, it is vital for BSs to

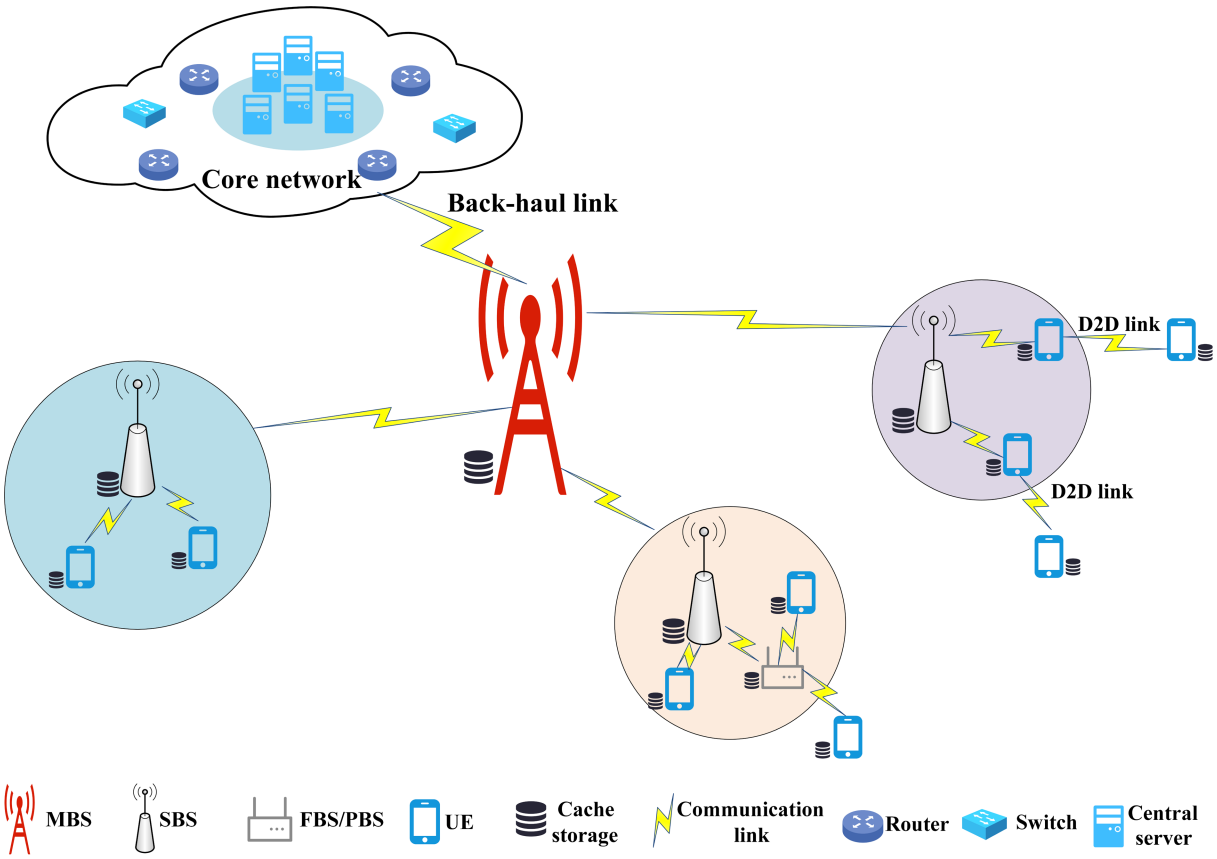

Fig. 4 A simplified architecture of edge caching network system.

Table 4 Configuration comparison of BSs[15].

BS		Application scenario	Maximum transmitting power (W)	Bandwidth (MHz)	Maximum coverage radius (m)	Number of users	Manufacturing cost
MBS		Outdoor	40–100	60–70	10 000–40 000	200–1000+	High
SBS	PBS	Indoor or outdoor	250	20	200	32–100	Medium
	FBS	Indoor	0.020–0.100	10	10–50	4–16	Low

alter their caching targets based on content popularity. Gu et al.[23] modelled the cache replacement process of BSs as a Markov Decision Process (MDP) and proposed a distributive scheme based on Reinforcement Learning (RL) algorithm, aimed at reducing transmission costs between different BSs. In 5G wireless networks, the total energy consumption and the backhaul constraint have emerged as critical factors impacting system performance and user QoE. Focused on these problems, Zhang et al.[25] proposed an HetNet architecture with the control-plane and user-plane, where the MBS and SBSs with different caching capabilities were stacked and collaborate. Results showed that the proposed architecture could improve the throughput and EE greatly. Likewise, Guo et al.[26] proposed a distributive BS dormant technique which took caching into account for enhancing HetNets' EE, while overcoming excessive backhaul expenditure. The power consumption in a system where an MBS shares with multiple SBSs was also considered in the proposed approach.

In cache-enabled HetNets, one of the key difficulties is determining how many cache items should be stored at different BSs in order to ensure assured service in a cost-effective way. Zhang et al.[27] designed a two-tier cache-enabled HetNet in a hierarchical structure, where the most popular instances were cached at SBSs, while the less popular ones at MBSs. The cache sizes of two types of BSs were optimized in order to increase network capacity and transmission rate. Furthermore, in Ref. [28], they investigated the cost-effective design of IoV made up of traditional MBSs and a special kind of SBSs, namely the cache-enabled Roadside Units (RSUs). Many factors were jointly optimized to reduce the deployment cost, under the constraints of QoE criteria and restricted backhaul capacity. In IoV environment, requests from fast-moving mobile users may affect SBSs' content popularity distribution. To avoid this, Li et al.[29] proposed a two-level architecture with multiple MBSs and SBSs, where high-speed users were always served by MBSs, while low-speed users were served by SBSs in the same cluster together.

Although the researches above have exploited the limited caching space, energy consumption, or the backhaul limitation, the security of cache-enabled HetNets is not considered. Focused on this point, Zheng et al.[30] studied the physical-layer safety for an HetNet consisting of one MBS and several SBSs. A joint design was proposed on caching placement and file transmission, aimed at realizing safe and energy-efficient delivery, confronted with randomly distributive listeners.

3.1.2 Caching at SBSs

Compared with MBS, SBS is featured by lower manufacturing cost and narrower coverage radius. Hence, it can be deployed more in an edge caching system, in collaboration with MBSs. With the paradigm transition from homogeneous networks to heterogeneous ones, the notion of caching at SBSs has gained great attention, which has the potential to support low power and high rate transmission. For example, in Ref. [31], Zhu et al. designed a hybrid architecture merging conventional SBSs and SBSs with caching capability, aimed at maximizing area SE while minimizing backhaul cost. They confirmed that in order to maximize SE, the most popular content should be cached, and a balance should be achieved between BS density and cache size.

In addition, many AI algorithms and mathematical theories have been applied to solve problems in SBS caching. For instance, Sengupta et al.[32] applied RL methods to study the distributive caching schemes in an HetNet composed of SBSs. To handle the joint cache placement optimization, which turns out to be NP-hard, a coded caching framework was proposed, where SBSs first learn the file popularity through a classic RL technique, and then pre-cache encoded fragments of popular files regularly, so as to better satisfy user requests. The game theory has also been utilized and combined with RL algoritms. For example, Hamidouche et al.[33] proposed a self-organizing RL algorithm modelled by the minority game theory, aimed at the managing backhaul of the 5G SBS HetNet. The algorithm designed was confirmed to

surpass the traditional greedy algorithm while not threatening requests' Quality of Service (QoS).

Since each SBS is just equipped with limited caching space, it is necessary for SBSs to adjust their caching targets based on the request record. Krishnendu et al.[34] presented a novel edge caching strategy for the SBS caching, which addressed the LP problem of maximizing average CHR in an approximate method. Results showed that the designed algorithm has a better performance compared with traditional greedy caching, Least Recently Used (LRU), and Least Frequently Used (LFU). Further, in Ref. [35], Krishnendu et al. proposed a caching strategy based on Federated Learning (FL), aimed at optimizing the CHR of the distributed SBS edge caching network. Unlike most previous efforts, which only take the static requirements into account, in their work, the data owned by each SBS were thought to be related spatiotemporally. And results showed that their algorithm performed better than recent online algorithms.

3.1.3 Caching at PBSs

As one kind of special SBS, PBS can be also applied in edge caching, which can also be referred to as pico-caching. Caching at MBS along with PBSs is a promising method to support massive content delivery and reduce backhaul cost in HetNets. many works have considered to utilize the pico-caching to assist the traditional MBS caching.

For example, Cui et al.[36] attempted to minimize the total time by optimizing caching strategy and considering user correlation, so as to meet the average user demands. The problem was modelled as a mixed discrete-continuous optimization under the constraints of bandwidth and caching space. Specifically, they demonstrated that the best policy is to store the most popular contents at each PBS. Furthermore, in Ref. [37], Jiang and Cui proposed a mixed two-tier caching design composed of an MBS-tier and a PBS-tier, and attempted to maximize the successful transmission probability of the HetNets, which can also come down to a mixed discrete-continuous optimization problem. An approximate solution to the problem was found, which had better performance and controllable complexity. There are also many other works focusing on the design of two-tier HetNet. For instance, Yi et al.[38] proposed a novel HetNet architecture, where MBSs were covered with intensive PBSs. The BSs in each layer were modelled as an homogeneous Poisson Point Processes (PPP) respectively, and cached contents based on content popularity ranking.

3.1.4 Caching at FBSs

An FBS, also known as a helper node or a home BS, is another kind of special SBS with smaller coverage and lower cost compared to FBS. Since FBSs are more flexible and cost-effective to deploy than conventional BSs, they are particularly suitable for indoor deployment, in order to improve indoor coverage which cannot be coverd by MBSs[39]. Since the concept of FBS was proposed, many studies have concentrated on caching at FBSs, which can be also referred to as femto-caching. Femto-caching can function as a supplement to other cache-enabled BSs, which has the great potential to improve the performance of edge caching.

For example, Lee and Lin[40] proposed a BS reselection caching strategy aimed at accelerating the transition of the HetNet structure from MBS-tier to FBS-tier. In Refs. [41, 42], Liu and Yang compared the SE of two different network structures, where the MBS-tier in the HetNet is overlaid by PBS-tier and FBS-tier, respectively. Results showed that in order to achieve the same target SE, the required density of FBS is lower than that of PBS, and by increasing increasing the caching space, the density can be further reduced. Considering a cache-enabled HetNet with FBSs spread around MBSs, Kuang and Liu[43] proposed a hybrid files caching strategy based on the request probability, where the most popular files are entirely cached in FBSs, while the less popular ones are partly cached. Using stochastic geometry theory, the locations of two different types of BSs were modeled, and an optimal strategy was designed to maximize the average SE.

3.2 Caching at end devices

End device generally refers to device which directly interacts with users, such as laptop and mobile phone, also known as User Equipment (UE). The proliferation of smartphones in the last decade has led to a sharp increase in data traffic, challenging the capacity of network infrastructure and mobile devices. To cope with this amount of traffic, Device-to-Device (D2D) communication has become a key feature for enhancing the performance of 5G cellular network[111, 112]. Current smartphones are becoming more complex with stronger computing and storage abilities. Therefore, mobile devices can serve as caching devices

themselves, caching locally and sharing with other terminals directly via D2D communication. Via D2D communication, adjacent end devices can skip the BSs to establish a direct link, thereby relieving the burden of network infrastructure and improving the network capacity dramatically. This type of caching is called D2D caching[44], which can bring contents closer to users and reduce backhaul traffic, thus optimizing users' QoE.

The growing demand for videos on mobile devices is a huge challenge to current network architecture. To address this challenge, Wu et al.[45] proposed a D2D caching based video transmission mechanism, which allows end users to cache and share videos cooperatively, aimed at improving mobile users' QoE. In response to the same demand, Anjum et al.[46] suggested a two-tier D2D caching approach aimed at reducing the delay experienced by Video-on-Demand (VoD) mobile users. In the proposed approach, caching capacity of the end device is split into two parts. The first part is dedicated to caching and delivering the initial portion of the most popular videos, while the second one is responsible for caching the remaining portion totally or partly in accordance with users' watching behavior and video popularity.

In Ref. [47], Panahi et al. pointed it out that the dense deployment of SBSs can suffer the EE of the HetNet, while the D2D caching can overcome this problem. In order to maximize EE, they introduced a D2D tier to the HetNet, along with saving energy consumption by hibernating some SBSs. Moreover, In Ref. [48], Meng et al. focused on the performance for the D2D caching network with Energy Harvesting (EH) function, where each end device can cache from BSs and get charged via EH. Stochastic geometry was used to model the network transmission, and then, two probabilistic caching schemes with different goals were proposed to jointly optimize CHR and successful offloading probability. In addition, the game theory has also been applied to optimize D2D caching. Shi et al.[49] pointed out that the selfish nature of users is the primary barrier to D2D caching, and the Stackelberg game was used to modeled the interest conflict between the operator and mobile users. Then, they proved that the game equilibrium exists, and designed an algorithm with low complexity to jointly optimize the incentive price and caching strategy approximately.

3.3 Cooperative BS/D2D caching

In a cache-enabled D2D cellular network, collaboration between BS caching and D2D caching can exploit the limited caching capacity adequately and achieve more efficient resource utilization, thus relieving core traffic and strengthening network capability dramatically. This caching strategy can be referred to as cooperative BS/D2D caching[50], where delivery traffic can be offloaded to end devices' cache through D2D link, or else directly to cache-enabled BSs through cellular link.

As to cooperative BS/D2D caching, one of the key questions is to decide whether popular contents should be cached at end devices or at BSs. In Ref. [51], Chen and Kountouris explored this problem by modelling the network using stochastic geometry and analyzing the performance of two different architectures (i.e., D2D caching and BS caching). Results showed that the performance depends on content popularity distribution and user density heavily. In Ref. [52], Jiang et al. proposed an optimal cooperative edge caching and delivery policy combining femto-caching and D2D caching. The cooperative problem was modelled as an Integer Linear Programming (ILP) problem, and a decomposition method was used to decompose the original problem into two sub-problems. In addition, considering the latency, it is important for the network to provide users with timely service responses. In Ref. [50], Soleimani and Tao proposed a strategy to evaluate whether a caching device can transmit the requested contents within the tolerant delay. Further, they designed a cooperation offloading strategy which integrated the transmission from BS caching and D2D caching to ensure the specified delay. Likewise, Wang et al.[53] also investigated the cooperation of BS caching and D2D caching, aimed at improving the successful transmission probability and guarantee the delay. A descent algorithm was proposed to resolve this collaborative caching placement problem.

4 Caching Object

As to the implication of the issue "what to cache", opinions vary among different researchers. On one hand, in Ref. [16], this issue is equated with the caching placement policies, which are designed to decide what content should be cached. In systems equipped with edge caching, such as cache-enabled HetNet, both the caching and backhaul capacity are finite. An idea to solve this problem is to equip BSs with larger caching capacity[113]. However, as a matter of fact, the caching capacity is always not enough

relative to massive amounts of contents in the network. For this reason, a proper caching placement policy is always indispensable in order for more appropriate contents to be stored in the edge cache, which will improve the performance metrics at both the cache-based and system aspects. On the other hand, both in Refs. [15, 18], the issue is interpreted as the classification of the caching content. The former classified the edge caching contents into three types: (1) time tolerant data; (2) time sensitive data; (3) IoT data, according to the analysis of users' content request types. The latter discussed the classification of edge caching contents for intelligent applications, and based on requests' redundancy, the caching contents were divided into two categories: (1) data redundancy content and (2) computation redundancy content.

Nevertheless, although the two works above have both tried to classify the edge caching contents, their classification patterns have certain one-sidedness and limitations. In this section, we will follow the latter comprehension of the issue "what to cache" in general, while attempting to propose a more novel and systematic classification method. In order to carry on a more comprehensive elaboration to this issue, having synthesized many works on edge caching, we tend to refer to the issue "what to cache" as the classification of the "caching objects". Specifically, the edge caching objects can be classified into three broad categories: (1) content cache; (2) data cache; (3) service cache, which will be discussed respectively in the following subsections, as shown in Table 5. Furthermore, the comparison of these three types of caching objects is shown graphically in Fig. 5.

4.1 Content cache

Content cache, such as files, videos, and webpages, is the most commonly cached object of edge caching, and is also the caching object of the CDN, which is one of the most mature application scenarios for edge caching.

The critical idea of in-network cache can be recognized as exploring and exploiting the spatiotemporal redundancy in user requests[54], and the same idea also holds true for edge caching techniques. The redundancy can largely determine the feasibility of caching techniques. According to Ref. [18], caching redundancy can be divided into two main categories,

Table 5 Classification of caching objects.

Type of caching objects		Application scenario	Reference
Content cache		Content (e.g., videos and webpages) transmission in CDN or IoT	[45, 54–61]
Data cache	Result cache	Storage of computation results from smart applications (e.g., image recognition and music identification)	[62–65]
	Self-data cache	Storage of monitoring data generated from devices (e.g., smart devices and IoT devices)	[66–68]
Service cache		Application services offloading at edge	[69–72]

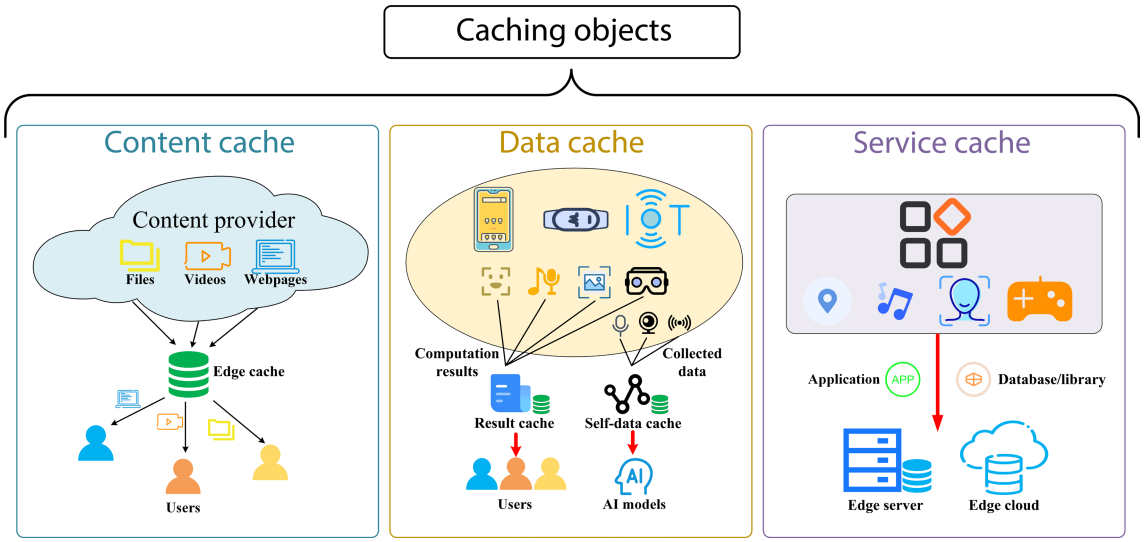

Fig. 5 Comparison of three types of edge caching objects.

i.e., communication redundancy and computation redundancy. The repeated request of popular multimedia contents can bring about communication redundancy. Since contents with high popularity tends to be requested by users frequently, as a result, the network has to delivery the duplicate contents over and over again. In this case, caching popular content at BSs or end devices can eliminate huge repetitive transmissions in the core network, thus relieving the strain on core network and improving users' QoE.

According to Cicso's latest report[1], video devices can have a huge effect on traffic especially. Because of the the increasing demand of users for high quality video experience, video devices will have an even more evident impact on the traffic. This throws up a huge challenge for current network. Therefore, to better design the content cache so as to meet the challenge, many existing studies have investigated users' request of various types of content, especially for videos. For example, Crane and Sornette[55] researched massive amounts of videos on YouTube in order to describe the activity of video requests. They found that the poisson process can be used to model most activity accurately, while other activities can be classified as three distinct patterns. Meanwhile, Traverso et al.[56] built a realistic arrival mechanism for content requests using access records from 60 000 YouTube end users. A new model was then proposed to allow users to capture the dynamics of content popularity locally and is effective for analysing caching systems. Based on these theoretical findings, many researchers have attempted to construct algorithms and mechanisms for caching video contents at edge. For instance, Tanzil et al.[57] applied ML method to construct an adaptive strategy for edge video content caching. The proposed strategy accounted for user pattern and network properties comprehensively, intended to select appropriate caching locations and caching sizes in the network. In Ref. [58], Sermpezis et al. investigated the process of caching popular content at FBSs and then introduced a new concept of "Soft Cache Hits" (SCHs), which extended the meaning of the traditional cache hits, considering using relative contents to compensate for cache misses. SCH is capable of complementing the service strategy of many applications like YouTube well, and it is shown that when SCHs are introduced, the optimal caching strategy should be redesigned. Moreover, in Ref. [45], Wu et al. designed a user-centric video transmission mechanism based on D2D communications allowing user video caching and sharing. The proposed mechanism considered various factors jointly, such as user relevance, user sharing willingness, etc.

The edge content caching has also played a vital role in IoT scenarios such as IoV. The development of the IoV has spawned a series of driving assistance services, which enhances the safety and intelligentization of transportation[59, 60]. As to self-driving, the edge computation offloading along with content caching is one of the key issues to offer high-quality vehicular services. For instance, in Ref. [61], Tian et al. suggested a collaborative Deep Reinforcement Learning (DRL) based scheme for compute offloading and content caching, in order to overcome the high mobility and dynamic requests of self-driving vehicles, so as to support informed offloading decision-making while assuring efficient utilization of content caching resources.

4.2 Data cache

Data cache is generally required by various emerging applications requesting a large amount of data all the time in order to accomplish certain tasks, e.g., various smart applications and IoT applications. These applications create massive volumes of monitoring, observation, and computation data, and requesting these data might spell enormous network traffic load[114]. Confronted with the growing demand of mobile users for higher service quality, in spite of the increasing computational power of smart mobile devices, they may still not be equal to those computationally intensive applications in a short period of time, like VR, AR, face recognition, etc. In addition, the huge power consumption involved with operating these applications remains a substantial barrier preventing users from fully enjoying these services. Activating devices frequently for data acquisition or calculation will drain their batteries and this is a major challenge. Therefore, it is beneficial for cache data for improve applications' efficiency, while saving network traffic and energy consumption.

In this paper, we classify the data cache into two main categories: (1) result cache; (2) self-data cache.

4.2.1 Result cache

In this AI era, we are now accustomed to a wide variety of intelligent edge devices, such as smart phones, laptops, etc. These intelligent edge devices

provide users with diverse smart applications based on AI model, in order to provide convenience and richness to users' daily life. For example, AI assistants with voice recognition, such as Siri, can communicate with mobile users naturally in real time, providing them with all kinds of services. Furthermore, users can turn to song identification module built into music applications for help to identify their interested songs, or use a plant picture recognition application to identify unknown flowers. Nevertheless, in spite of the great convenience these smart devices have brought to our lives, frequent use of these AI-based applications with high computational complexity may spell computation redundancy, which can cause great power consumption of devices[62, 63].

On the other hand, many researchers have realized the computation redundancy presence in applications with AI techniques, and attempted to eliminate it. For example, when walking in a park, many visitors may perform plant recognition on the same flowers. In this case, without the data cache, when each user tries to identify the same flower, their edge device will perform the exact same task all over again. In this process, there exists lots of unwarranted computations across devices. However, if such an image recognition task can be offloaded to edges, and the recognition results can be cached, redundant computations can be further eliminated[64], which is exactly the core idea of the result cache, which belongs to edge data cache. By caching the correspondence between specific tasks and results, along with the assistance of D2D communication among different users, the execution of repetitive computational tasks can be reduced to a certain extent, thus optimizing the users' experience of the applications while reducing the devices' power consumption from activating tasks with high computational complexity frequently.

The same application is often called on plural devices in close distance, and it often process similar inputs which will correspond to the same result. Based on this pattern, Guo et al.[64] exploited the equivalence between different application inputs in order to reuse the results of previous calculations, aimed at minimizing redundant computations. This was an early attempt to carve out and eliminate the computation redundancy of application input data by using real datasets. In addition, Xia et al.[65] utilized collaborative cache-enabled BSs to guarantee low latency users' retrieval of application data. They then designed an online algorithm to deal with caching resource limitations, which was proved to obtain the approximate optimal performance.

As to computation redundancy and result cache, the key step is to mine and measure the similarity of the features of users' requests. Then, we need to find the best mapping from input features to computation results. It is worth emphasizing that from the perspective of computation redundancy, the caching object is computation results of smart applications, instead of users' requested files in content cache (i.e., communication redundancy), which is where the main difference between the two lies.

4.2.2 Self-data cache

When smart applications work, edge devices may be activated to monitor the surrounding environment. For example, when a picture recognition application is used, the device's camera lens may need to be invoked to collect the picture. Additionally, when a music app's song identification module is called, the device's audio recording component requires to be activated. Another typical example is for IoT applications. Since the operation of IoT applications requires massive amounts of monitoring information about the surroundings, IoT devices must be activated periodically to obtain data. However, frequent activation of devices to fetch data may increase power consumption and reduce service life in the long term.

Unlike the result cache, this kind of data cache discussed here refers to the data generated by devices themselves or acquired from the surroundings, which can be called the self-data cache. Especially, it is beneficial for IoT devices to cache IoT data so as to reduce the activation frequency[66]. For one hand, since the much shorter lifespan of IoT data, without the self-data cache, IoT devices will have to be activated now and again. On the other hand, to ensure the timeliness and availability of the collected data, it is necessary to evaluate the freshness of the cached data, and then update in due course. Therefore, more intelligent caching strategies are desired to strike a balance between data freshness and caching frequency. For instance, Vural et al.[67] first applied in-network caching techniques to IoT data. They investigated the IoT data caching at content routers and modelled the trade-off between data freshness and communication costs. Likewise, in Ref. [68], Zhu et al. applied DRL

algorithm to solve the problem of edge IoT data caching, which considered the variability of both the IoT data and IoT environment. By designing metrics to measure data freshness, they found that the key to caching IoT data is to achieve a trade-off between communication costs and data freshness loss, which is consistent with the view put forward in Ref. [67].

4.3 Service cache

Service cache refers to the caching of application services and the associated databases/libraries at BS or edge cloud[115–118], so that the corresponding tasks can be executed at the edge[69]. Whether for content cache or data cache, the cached object will only occupy the storage resources of the caching devices. However, for the service cache, since the cached service needs to be executed on edge devices, apart from storage resources, computing resources also require consideration, which is where the hugest difference between service cache and previous two kinds of cache lies. Compared with content cache and data cache, service cache is an equally important kind of caching object worth studying, yet receives much less attention.

Unlike caching services at cloud with huge and diverse resources, when considering caching service at edge, the limitation of caching devices on both computing and storage resources is a huge challenge. As a result, as to caching services at edge, only a small quantity of services can be cached simultaneously at edge with limited resources, and judicious decisions must be mapped out about which services to cache in order to improve the performance of EC.

To overcome this issue, some existing researches have attempted to propose some algorithm or strategy to optimize the edge service caching. For example, focused on joint dynamic service caching and assignment offloading, Xu et al.[69] devised a Lyapunov optimization and Gibbs sampling based online algorithm, taking into account the complexity of both service requests and environment. In addition, Huang and Shen[70] proposed a novel service caching architecture to cache popular services in edge clouds autonomously. Along with the architecture, a new caching replacement strategy was also designed to maximize the service CHR, which was verified superior to other current strategies in improving CHR and reducing average delay. Moreover, Zhang et al.[71] proposed a collaborative optimization method to tackle the Mixed-Integer Nonlinear Programming (MINLP) problem aimed at high-efficient service caching resource distribution and computation offloading. Similarly, Tran et al.[72] modelled the joint optimization of service caching costs and power consumption as a Mixed-Integer Linear Programming (MILP) problem, and then designed an iterative algorithm to find an acceptable approximate optimal solution efficiently.

5 Caching Strategy

Since the concept of caching was introduced into computer science, it has become a core issue in caching to design a reasonable and effective caching strategy[119]. The purpose of caching is to enhance the speed of retrieval by storing data selectively. Therefore, it is necessary to design a sensible caching strategy so that caching can achieve its desired effect. Caching strategy is a large category in itself, which contains numerous “sides”.

In this section, the edge caching strategies will be compared from two distinct perspectives: (1) reactive caching vs. proactive caching; (2) centralized caching vs. distributed caching. The comparison of these two pairs of caching strategies is shown in Table 6. Conventional web caching usually adopts a reactive and centralized caching strategy, which is simpler while insufficient to meet specific network application environments, making it difficult to demonstrate the merits of caching. In virtue of the distributive nature of EC, many existing studies have attempted to innovate algorithms on conventional caching strategies or to design novel edge caching strategies, which will be discussed in the following subsections.

5.1 Reactive caching vs. proactive caching

The edge caching placement strategy can be categorized as reactive caching strategy and proactive caching strategy, depending on whether caching or replacement behavior occurs only at the arrival time of the request or in advance by using network information. The key difference between the two strategies lies in the precedence of cache updates and user requests. The former determines whether to cache a particular object after the arrival of a user request, while the latter determines which objects to cache before they are requested based on the prediction of user requests, which is also called the pre-caching technology. The comparison between the process of reactive caching and proactive caching is shown in Fig. 6.

Table 6 Comparison of caching strategies.

Strategy	Characteristic	Key issue	Reference
Reactive caching	Determine whether to cache a particular object after the arrival of a user request.	Constructing novel caching replacement algorithms; Introducing novel caching metrics	[73–77]
Proactive caching	Determines which objects to cache before they are requested based on the prediction of user requests.	Object popularity estimation; User patterns prediction	[78–85]
Centralized caching	Be centered on the central controller which can check the global state of the network to make appropriate caching decisions.	— (Unsuitable for edge caching)	[86–88]
Distributed caching	Caching devices make their own decisions (e.g., caching placement and replacement) according to the local information or the information from their neighboring nodes, instead of following the instructions from the central controller.	Designing effective and efficient distributed caching algorithms	[89–92]

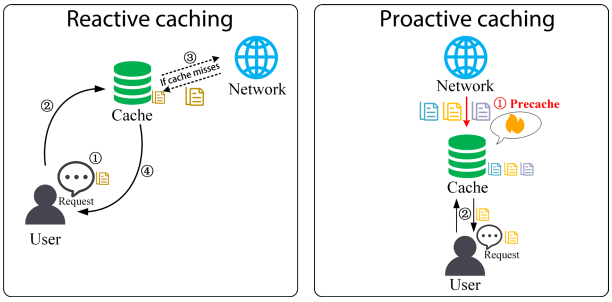

Fig. 6 Comparison between the process of reactive caching and proactive caching.

Traditional caching strategies used to be reactive, the key to which is various caching replacement algorithms, such as First In First Out (FIFO), LRU, and LFU[120]. On the other hand, with the aid of AI technology, it is advantageous to estimate users' request patterns and cache some popular objects in advance.

5.1.1 Reactive edge caching strategy

As to reactive edge caching strategy, some existing works have attempted to combine several conventional caching replacement algorithms to construct new caching strategy, or to introduce new metrics into conventional caching strategies to form novel reactive caching strategies.

For one thing, some reactive caching strategies incorporating different algorithms can get better performance in edge caching. For example, based on the second-chance concept, Alghazo et al.[73] designed a new caching replacement algorithm called Second chance-Frequency-LRU (SF-LRU), which combined LFU and LRU together. Results showed that the SF-LRU algorithm can improve CHR significantly compared with both LRU and LFU. Further, in Ref. [74], Subramanian et al. proposed an adaptive reactive caching strategy which is able to combine any two traditional replacement algorithms and switch between them according to the running state.

For another, some new metrics have been designed from EC and added into conventional caching strategies so as to generate novel reactive caching strategies. For instance, Hu and Johnson[75] established a series of new mobility metrics to accurately measure the relative complexity that a given movement scenario brings to the caching strategies in a specific routing protocol. Dimokas et al.[76] introduced two novel metrics to assist the caching location selection and then proposed two novel cooperative caching strategies based on the metrics. Results showed that the designed strategies can achieve lower delay and higher CHR against their opponent, NICoCa. Moreover, in Ref. [77], Yeh et al. proposed a new metric Virtual Interest Packets (VIP) to measure the demand for data objects in the network. Then, a novel framework based on VIP metric was designed to achieve the load balance via dynamic delivering and caching, aimed at reducing latency and improving CHR.

5.1.2 Proactive edge caching strategy

Human behavior is closely related and divinable[78], which has great enlightenment significance for the proposal and design of the proactive edge caching strategy. With the assistance of AI and big data analytic technology, it is feasible to predict users' demand and pre-cache popular objects locally before the true arrival of the requests[79–81]. In general, as to proactive caching strategy, caching objects are cached actively at BSs or end devices during off-peak periods according to user demand prediction according to the popularity and relevance of user patterns. Then, during peak periods, traffic load can be observably relieved

via these pre-cached objects[82], thus improving the caching efficiency immensely compared with reactive caching strategy.

In order to confirm the effectiveness of proactive edge caching, Bastug et al.[82] investigated two cases where proactive caching plays a critical role. Through the detailed analysis of proactive caching paradigm, results showed that the introduction of proactive caching strategy can deliver significant gains for edge caching, saving the backhaul costs and improving the user satisfaction rate considerably. They also pointed it out that this kind of gains can even be higher with more sufficient caching space.

In Ref. [83], Zeydan et al. proposed a 5G architecture combining proactive caching and big data technique. By using ML tools, a mass of data were utilized for object popularity estimation, and then valuable objects were pre-cached at BSs based on prediction, which was shown to enhance user satisfaction and reduce backhaul cost. Moreover, Ale et al.[84] proposed an online deep learning based proactive caching strategy which can predict sequential user requests and update edge caching correspondingly, so as to improve the accuracy of popularity prediction and then improve the CHR of end devices. The game theory is also applied to design proactive edge caching strategy. In Ref. [85], Zheng et al. modelled the proactive edge caching process as a Stackelberg game, which can be further broken up into two types of sub-games. Then they designed a distributed algorithm to solve the sub-games, which turned out to be linearly or sub-linearly related to the network scale, showing that the strategy has the potential to cope with large-scale edge caching.

5.2 Centralized caching vs. distributed caching

Centralized caching takes the central controller as center, which can check the global state of the network to make appropriate caching decisions. The central controller will monitor network state and user patterns at any time through analysis of received requests in order to make caching policies. Hence, the goal of centralized caching is to optimize the caching performance of the entire system globally by optimizing caching decisions. Nevertheless, in the wave of 5G, which is projected to service a growing number of mobile users, it is especially challenging for the centralized caching to obtain global network information due to the high dynamics of mobile networks[86, 87]. Furthermore, in the centralized caching, the central controller has to handle large amounts of traffic, which will pose a tremendous burden on both itself and the network links. Therefore, the centralized caching strategy has become a bottleneck affecting the performance of mobile caching systems[88].

On the contrary, as to distributed caching, also known as decentralized caching, caching devices will make their own decisions (e.g., caching placement and replacement) according to the local information or the information from their neighboring nodes, instead of following the instructions from the central controller. The comparison between the characteristic of centralized caching and distributed caching is shown in Fig. 7. EC is a distributed model, which decentralizes computation and storage tasks to the edge closer to users. From this perspective, the distributed caching strategy can make better use of caching devices like BSs and end devices in the edge caching system to participate in caching decision-making, thus improving the caching performance of the whole system. Therefore, the distributed caching is a more appropriate caching strategy for edge caching, which matches the characteristics of EC.

The key to the distributed caching strategy is to design effective and efficient distributed caching algorithms, in order to solve the caching optimization problem in a distributed system. For example, in Ref. [89], Borst et al. investigated the edge caching for video content delivery and proposed a distributed and cooperative caching management algorithm aimed at maximizing CHR and minimizing the bandwidth overhead. They focused on distributed caching clusters and modelled the content caching placement as a Linear Programming (LP) problem in order to obtain the global optimal solution, which provided meaningful

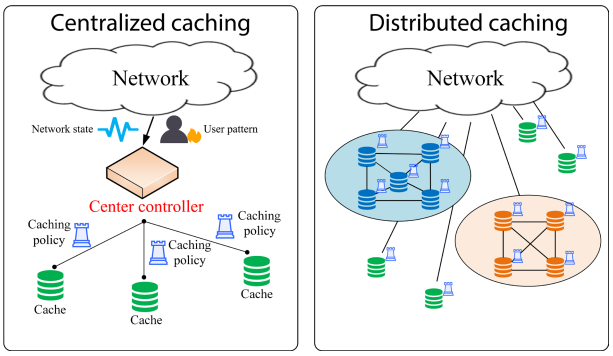

Fig. 7 Comparison between the characteristic of centralized caching and distributed caching.

inspiration for designing caching algorithms with low complexity. Likewise, focused on optimizing VoD transmission in wireless network, Shanmugam et al.[90] designed an edge caching system with MBSs and auxiliary FBSs to cache popular video files distributively, aimed at maximizing video transmission efficiency. In order to explore the optimal femto-caching placement algorithm for video files, they distinguished the cases into the uncoded method and coded method, and confirmed that the latter method holds more advantages. In addition, in Ref. [91], Liu et al. designed a distributed edge caching algorithm based on probability propagation algorithm, aimed at minimizing the average download delay. In their designed algorithm, each BS can compute and make caching decisions parallelly based on finite local information and very few information from adjcent ones, thus requiring no central controller to collect the global information and saving backhaul cost. Concentrated on distributed edge caching for IoT devices, Tian et al.[92] proposed a novel distributed micro-service caching strategy. They modelled the caching problem as an MDP, and then designed an RL-based algorithm to optimize the delay and CHR, which allowed each IoT device to make caching decisions independently and distributively.

6 Challenge and Open Issue

Although many existing studies have explored the edge caching techniques from various aspects and made substantial gains, there still remains many challenges and open issues to be discovered and addressed in this topic, which will be briefly discussed below in this section, including privacy and security, intelligent edge caching, mobile edge caching, and IoT edge caching.

6.1 Privacy and security

By utilizing the computing and storage resources at the edge of the network, edge caching can reduce network delay, relieve network burden, and enhance users' QoE significantly. However, in spite of these advantages, edge caching is subject to lots of threats regarding privacy attacks and security hazards[121, 122]. For one thing, the proactive edge caching strategy requires a large number of user data to optimize the network caching performance, which may contain sensitive privacy information and lead to user privacy exposure. For another, the edge caching strategy is distributed generally, which makes the edge caching system more susceptible to various types of security attacks[123].

The security and privacy have always been an open issue to be solved in EC, and the same is true for edge caching. In recent years, some researchers have attempted to leverage AI techniques to overcome the potential privacy and security issues of edge caching. Typically, federated learning is a distributed ML framework with the advantage of protecting privacy and guaranteeing data security[124], and thus is often applied in privacy and security protection for edge caching. For example, Yu et al.[125] proposed a proactive edge caching strategy based on FL, in order to improve the CHR in IoV edge caching, while protecting user privacy effectually. In addition, the RL technique is also another powerful tool to deal with this problem. For instance, in Ref. [123], Xiao et al. applied RL method to protect the mobile edge caching from security attack and privacy disclosure. Similarly, in Ref. [126], Liu et al. proposed a distributed RL-based algorithm to maximize the CHR in the Mobile Edge Computing (MEC) networks while considering the constraint conditions of the privacy protection.

6.2 Intelligent edge caching

In the 5G wireless networks, the network should be more adaptive and intelligent to offer better user experience to an increasing number of mobile users. As the structure of the network becomes progressively more complex, and the user demands gradually increase, traditional caching strategies are no longer capable and applicable to edge caching. Therefore, AI techniques must be combined with edge caching so that the designed caching system can collect network information more intelligently, make smarter decisions, and better serve users.

Intelligent edge caching has always been an issue of enduring vitality and endless potential in the field of edge caching[127]. Apart from the presented efforts to edge caching system design using AI techniques in this paper, with the development of AI technology, intelligent edge caching still has a long way to go to propose more intelligent, more efficient, and better-performing edge caching systems.

6.3 Mobile edge caching

With the development of 5G technology along with the widespread employment of numerous mobile applications, the MEC has gained great attention, which brings computing and storage resources to the edge of mobile networks. The mobile edge caching can

cache content, data, or services at BSs or end devices, and transmit them by utilizing wireless communication, thus improving the QoE of mobile end users. As one of the three major scenarios for 5G, Ultra-Reliable and Low-Latency Communication (URLLC)[128] requires the network to be capable of offering high-quality services with extremely low latency to many applications, such as online games, AR, VR, etc.

The issue of technical applications for mobile networks remains a hot topic for future research. Therefore, the mobile edge caching is one of the hottest research issues in edge caching at present and even in the future. The fundamental challenge of mobile edge caching is how to design effective strategies that can cope with the dynamic and volatile properties of mobile networks, so that the storage and computing resources of distributed edge devices can be fully utilized to optimize caching performance and enhance user experience. While complex caching strategies have the capability to improve users' QoE, they can be inefficient and take up too many computation resources. Thus, although many existing studies have proposed some intelligent mobile edge caching algorithms, it remains an open issue to design an edge caching algorithm capable of achieving the same effect, but with lower complexity.

6.4 IoT edge caching

Since the concept of the IoT was first proposed in 1999, in recent years, it has brought earth-shaking changes to our daily lives. The core of the IoT is to connect numerous devices with sensing capabilities, which can monitor the surroundings in real time, and utilize the acquired data to provide convenient services for human production and life[129, 130]. However, the high latency and high power consumption are the main barriers currently that prevent IoT facilities from reaching their true potential, while the consideration of introducing edge caching into IoT architecture can alleviate these problems effectively[131].

In fact, as previously illustrated, both the edge data caching and edge service caching are of profound significance to IoT networks, powering up Industry 4.0[132, 133]. In order to enable edge caching in IoT networks, designing edge caching strategies applicable to IoT is of great value. Nevertheless, current exploration of IoT edge caching is still inadequate, and more works are urgently required in this subdomain. Recently, some works have proposed to apply Unmanned Aerial Vehicles (UAVs) to assist edge caching in IoT. For instance, in Ref. [134], Gu et al. designed a unique kind of mobile edge caching system for IoT consisting of UAVs and satellites, aimed at optimizing the delay and energy consumption in IoT networks. In summary, the IoT edge caching is still an area worth exploring.

7 Conclusion

This article provides a clear survey of the present edge caching techniques comprehensively and systematically. The key issues regarding edge caching can fall into three main categories, i.e., where, what, and how to cache, which correspond to caching locations, caching objects, and caching strategies, respectively. The most commonly applied performance metrics of edge caching are introduced from the perspective of the cache itself and the entire network caching system, before the detailed elaboration of three key issues in edge caching in turn. In particular, the issue of "what to cache" is reinterpreted as the classification of caching objects from a novel perspective, which can be further divided into content cache, data cache, and service cache. At last, we also discuss some challenges and open issues about the edge caching from four different aspects. The aim of this survey paper is to summarize existing edge caching technologies systematically and to provide certain reference for edge caching design in the context of 5G and beyond 5G. According to the survey, there are still broad prospects for edge caching worth further researching, and there is still a long way to go to bring the true power of edge caching to bear.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 92267104), the Natural Science Foundation of Jiangsu Province of China (No. BK20211284), and Financial and Science Technology Plan Project of Xinjiang Production and Construction Corps (No. 2020DB005).

References

[1] U. Cisco, Cisco annual internet report (2018–2023) white paper, https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, 2020.

[2] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking, A review on Internet of Things (IoT), Internet of everything (IoE) and Internet of nano things (IoNT), in *Proc. 2015*

Internet Technologies and Applications (*ITA*), Wrexham, UK, 2015, pp. 219–224.

[3] X. Zhou, W. Liang, K. Yan, W. Li, K. I. K. Wang, J. Ma, and Q. Jin, Edge-enabled two-stage scheduling based on deep reinforcement learning for Internet of everything, *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3295–3304, 2023.

[4] J. Hendler and J. Golbeck, Metcalfe's law, web 2.0, and the semantic web, *J. Web Semant.*, vol. 6, no. 1, pp. 14–20, 2008.

[5] M. Alioto, Enabling the Internet of Things: From integrated circuits to integrated systems, https://api.semanticscholar.org/CorpusID:115720772, 2017.

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

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

[8] M. A. Maddah-Ali and U. Niesen, Fundamental limits of caching, *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[9] C. Aggarwal, J. L. Wolf, and P. S. Yu, Caching on the world wide web, *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 1, pp. 94–107, 1999.

[10] X. Sun and N. Ansari, Dynamic resource caching in the IoT application layer for smart cities, *IEEE Internet Things J.*, vol. 5, no. 2, pp. 606–613, 2018.

[11] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, Cache in the air: Exploiting content caching and delivery techniques for 5G systems, *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, 2014.

[12] M. Satyanarayanan, The emergence of edge computing, *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

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

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

[15] J. Yao, T. Han, and N. Ansari, On mobile edge caching, *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2525–2553, 2019.

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

[17] S. Safavat, N. N. Sapavath, and D. B. Rawat, Recent advances in mobile edge computing and content caching, *Digit. Commun. Netw.*, vol. 6, no. 2, pp. 189–194, 2020.

[18] D. Xu, T. Li, Y. Li, X. Su, S. Tarkoma, T. Jiang, J. Crowcroft, and P. Hui, Edge intelligence: Architectures, challenges, and applications, arXiv preprint arXiv: 2003.12172, 2020.

[19] H. Wu, Y. Fan, Y. Wang, H. Ma, and L. Xing, A comprehensive review on edge caching from the perspective of total process: Placement, policy and delivery, *Sensors*, vol. 21, no. 15, p. 5033, 2021.

[20] J. Shuja, K. Bilal, W. Alasmary, H. Sinky, and E. Alanazi, Applying machine learning techniques for caching in next-generation edge networks: A comprehensive survey, *J. Netw. Comput. Appl.*, vol. 181, p. 103005, 2021.

[21] M. Amadeo, C. Campolo, G. Ruggeri, and A. Molinaro, Edge caching in IoT smart environments: Benefits, challenges, and research perspectives toward 6G, in *IoT Edge Solutions for Cognitive Buildings*, F. Cicirelli, A. Guerrieri, A. Vinci, and G. Spezzano, eds. Cham, Switzerland: Springer, 2022, pp. 53–73.

[22] M. Reiss-Mirzaei, M. Ghobaei-Arani, and L. Esmaeili, A review on the edge caching mechanisms in the mobile edge computing: A social-aware perspective, *Internet Things*, vol. 22, p. 100690, 2023.

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

[24] J. Wen, K. Huang, S. Yang, and V. O. K. Li, Cache-enabled heterogeneous cellular networks: Optimal tier-level content placement, *IEEE Trans. Wirel. Commun.*, vol. 16, no. 9, pp. 5939–5952, 2017.

[25] J. Zhang, X. Zhang, and W. Wang, Cache-enabled software defined heterogeneous networks for green and flexible 5G networks, *IEEE Access*, vol. 4, pp. 3591–3604, 2016.

[26] W. Guo, S. A. Wagan, D. R. Shin, and N. M. F. Qureshi, Cache-based green distributed cell dormancy technique for dense heterogeneous networks, *Comput. Commun.*, vol. 191, pp. 69–77, 2022.

[27] S. Zhang, N. Zhang, P. Yang, and X. Shen, Cost-effective cache deployment in mobile heterogeneous networks, *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11264–11276, 2017.

[28] S. Zhang, N. Zhang, X. Fang, P. Yang, and X. S. Shen, Cost-effective vehicular network planning with cache-enabled green roadside units, in *Proc. 2017 IEEE Int. Conf. Communications* (*ICC*), Paris, France, 2017, pp. 1–6.

[29] L. Li, C. F. Kwong, Q. Liu, P. Kar, and S. P. Ardakani, A novel cooperative cache policy for wireless networks, *Wirel. Commun. Mob. Comput.*, vol. 2021, pp. 1–18, 2021.

[30] T. X. Zheng, H. M. Wang, and J. Yuan, Secure and energy-efficient transmissions in cache-enabled heterogeneous cellular networks: Performance analysis and optimization, *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5554–5567, 2018.

[31] Y. Zhu, G. Zheng, K. K. Wong, S. Jin, and S. Lambotharan, Performance analysis of cache-enabled

millimeter wave small cell networks, *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6695–6699, 2018.

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

[33] K. Hamidouche, W. Saad, M. Debbah, J. B. Song, and C. S. Hong, The 5G cellular backhaul management dilemma: To cache or to serve, *IEEE Trans. Wirel. Commun.*, vol. 16, no. 8, pp. 4866–4879, 2017.

[34] S. Krishnendu, B. N. Bharath, and V. Bhatia, Cache enabled cellular network: Algorithm for cache placement and guarantees, *IEEE Wirel. Commun. Lett.*, vol. 8, no. 6, pp. 1550–1554, 2019.

[35] S. Krishnendu, B. N. Bharath, N. Garg, V. Bhatia, and T. Ratnarajah, Learning to cache: Federated caching in a cellular network with correlated demands, *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1653–1665, 2022.

[36] Y. Cui, F. Lai, S. Hanly, and P. Whiting, Optimal caching and user association in cache-enabled heterogeneous wireless networks, in *Proc. 2016 IEEE Global Communications Conf.* (*GLOBECOM*), Washington, DC, USA, 2016, pp. 1–6.

[37] D. Jiang and Y. Cui, Caching and multicasting in large-scale cache-enabled heterogeneous wireless networks, in *Proc. 2016 IEEE Global Communications Conf.* (*GLOBECOM*), Washington, DC, USA, 2016, pp. 1–7.

[38] W. Yi, Y. Liu, and A. Nallanathan, Cache-enabled HetNets with millimeter wave small cells, *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5497–5511, 2018.

[39] V. Chandrasekhar, J. G. Andrews, and A. Gatherer, Femtocell networks: A survey, *IEEE Commun. Mag.*, vol. 46, no. 9, pp. 59–67, 2008.

[40] H. Y. Lee and Y. B. Lin, A cache scheme for femtocell reselection, *IEEE Commun. Lett.*, vol. 14, no. 1, pp. 27–29, 2010.

[41] D. Liu and C. Yang, Cache-enabled heterogeneous cellular networks: Comparison and tradeoffs, in *Proc. 2016 IEEE Int. Conf. Communications* (*ICC*), Kuala Lumpur, Malaysia, 2016, pp. 1–6.

[42] D. Liu and C. Yang, Caching policy toward maximal success probability and area spectral efficiency of cache-enabled HetNets, *IEEE Trans. Commun.*, vol. 65, no. 6, pp. 2699–2714, 2017.

[43] S. Kuang and N. Liu, Cache-enabled base station cooperation for heterogeneous cellular network with dependence, in *Proc. 2017 IEEE Wireless Communications and Networking Conf.* (*WCNC*), San Francisco, CA, USA, 2017, pp. 1–6.

[44] T. Wang, Y. Wang, X. Wang, and Y. Cao, A detailed review of D2D cache in helper selection, *World Wide Web*, vol. 23, no. 4, pp. 2407–2428, 2020.

[45] D. Wu, Q. Liu, H. Wang, Q. Yang, and R. Wang, Cache less for more: Exploiting cooperative video caching and delivery in D2D communications, *IEEE Trans. Multimed.*, vol. 21, no. 7, pp. 1788–1798, 2019.

[46] N. Anjum, Z. Yang, I. Khan, M. Kiran, F. Wu, K. Rabie, and S. M. Bahaei, Efficient algorithms for cache-throughput analysis in cellular-D2D 5G networks, *Comput. Mater. Continua*, vol. 67, no. 2, pp. 1759–1780, 2021.

[47] F. H. Panahi, F. H. Panahi, and T. Ohtsuki, Energy efficiency analysis in cache-enabled D2D-aided heterogeneous cellular networks, *IEEE Access*, vol. 8, pp. 19540–19554, 2020.

[48] Y. Meng, Z. Zhang, and Y. Huang, Cache- and energy harvesting-enabled D2D cellular network: Modeling, analysis and optimization, *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 2, pp. 703–713, 2021.

[49] L. Shi, L. Zhao, G. Zheng, Z. Han, and Y. Ye, Incentive design for cache-enabled D2D underlaid cellular networks using stackelberg game, *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 765–779, 2019.

[50] S. Soleimani and X. Tao, Cooperative crossing cache placement in cache-enabled device to device-aided cellular networks, *Appl. Sci.*, vol. 8, no. 9, p. 1578, 2018.

[51] Z. Chen and M. Kountouris, D2D caching vs. small cell caching: Where to cache content in a wireless network? in *Proc. 2016 IEEE 17th Int. Workshop on Signal Processing Advances in Wireless Communications* (*SPAWC*), Edinburgh, UK, 2016, pp. 1–6.

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

[53] Y. Wang, X. Tao, X. Zhang, and Y. Gu, Cooperative caching placement in cache-enabled D2D underlaid cellular network, *IEEE Commun. Lett.*, vol. 21, no. 5, pp. 1151–1154, 2017.

[54] I. Psaras, W. K. Chai, and G. Pavlou, In-network cache management and resource allocation for information-centric networks, *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2920–2931, 2014.

[55] R. Crane and D. Sornette, Robust dynamic classes revealed by measuring the response function of a social system, *Proc. Natl. Acad. Sci. USA*, vol. 105, no. 41, pp. 15649–15653, 2008.

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

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

[58] P. Sermpezis, T. Giannakas, T. Spyropoulos, and L. Vigneri, Soft cache hits: Improving performance through recommendation and delivery of related content, *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1300–1313, 2018.

[59] X. Xu, Q. Jiang, P. Zhang, X. Cao, M. R. Khosravi, L. T. Alex, L. Qi, and W. Dou, Game theory for distributed IoV task offloading with fuzzy neural network in edge

computing, *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 11, pp. 4593–4604, 2022.

[60] C. M. Martinez, M. Heucke, F. Y. Wang, B. Gao, and D. Cao, Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey, *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 666–676, 2018.

[61] H. Tian, X. Xu, L. Qi, X. Zhang, W. Dou, S. Yu, and Q. Ni, CoPace: Edge computation offloading and caching for self-driving with deep reinforcement learning, *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13281–13293, 2021.

[62] T. Braud, P. Zhou, J. Kangasharju, and P. Hui, Multipath computation offloading for mobile augmented reality, in *Proc. 2020 IEEE Int. Conf. Pervasive Computing and Communications* (*PerCom*), Austin, TX, USA, 2020, pp. 1–10.

[63] W. Zhang, B. Han, and P. Hui, Low latency mobile augmented reality with flexible tracking, in *Proc. 24th Annual Int. Conf. Mobile Computing and Networking*, New Delhi, India, 2018, pp. 829–831.

[64] P. Guo, B. Hu, R. Li, and W. Hu, FoggyCache: Cross-device approximate computation reuse, in *Proc. 24th Annual Int. Conf. Mobile Computing and Networking*, New Delhi, India, 2018, pp. 19–34.

[65] X. Xia, F. Chen, Q. He, J. Grundy, M. Abdelrazek, and H. Jin, Online collaborative data caching in edge computing, *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 2, pp. 281–294, 2021.

[66] X. Wang, S. Leng, and K. Yang, Social-aware edge caching in fog radio access networks, *IEEE Access*, vol. 5, pp. 8492–8501, 2017.

[67] S. Vural, P. Navaratnam, N. Wang, C. Wang, L. Dong, and R. Tafazolli, In-network caching of internet-of-things data, in *Proc. 2014 IEEE Int. Conf. Communications* (*ICC*), Sydney, Australia, 2014, pp. 3185–3190.

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

[69] J. Xu, L. Chen, and P. Zhou, Joint service caching and task offloading for mobile edge computing in dense networks, in *Proc. IEEE INFOCOM 2018 - IEEE Conf. Computer Communications*, Honolulu, HI, USA, 2018, pp. 207–215.

[70] C. K. Huang and S. H. Shen, Enabling service cache in edge clouds, *ACM Trans. Internet Things*, vol. 2, no. 3, pp. 1–24, 2021.

[71] Z. Zhang, H. Zhou, and D. Li, Joint optimization of multi-user computing offloading and service caching in mobile edge computing, in *Proc. 2021 IEEE/ACM 29th Int. Symp. on Quality of Service* (*IWQOS*), Tokyo, Japan, 2021, pp. 1–2.

[72] T. X. Tran, K. Chan, and D. Pompili, COSTA: Cost-aware service caching and task offloading assignment in mobile-edge computing, in *Proc. 2019 16th Annual IEEE Int. Conf. Sensing, Communication, and Networking* (*SECON*), Boston, MA, USA, 2019, pp. 1–9.

[73] J. Alghazo, A. Akaaboune, and N. Botros, SF-LRU cache replacement algorithm, in *Proc. Records of the 2004 Int. Workshop on Memory Technology*, *Design and Testing*, San Jose, CA, USA, 2004, pp. 19–24.

[74] R. Subramanian, Y. Smaragdakis, and G. H. Loh, Adaptive caches: Effective shaping of cache behavior to workloads, in *Proc. 2006 39th Annual IEEE/ACM Int. Symp. on Microarchitecture* (*MICRO'06*), Orlando, FL, USA, 2006, pp. 385–396.

[75] Y. C. Hu and D. B. Johnson, Caching strategies in on-demand routing protocols for wireless ad hoc networks, in *Proc. 6th annual Int. Conf. Mobile computing and networking*, Boston, MA, USA, 2000, pp. 231–242.

[76] N. Dimokas, D. Katsaros, L. Tassiulas, and Y. Manolopoulos, High performance, low complexity cooperative caching for wireless sensor networks, in *Proc. 2009 IEEE Int. Symp. on a World of Wireless, Mobile and Multimedia Networks & Workshops*, Kos, Greece, 2009, pp. 1–9.

[77] E. Yeh, T. Ho, Y. Cui, M. Burd, R. Liu, and D. Leong, VIP: A framework for joint dynamic forwarding and caching in named data networks, in *Proc. 1st ACM Conf. Information-Centric Networking*, Paris, France, 2014, pp. 117–126.

[78] M. Kosinski, D. Stillwell, and T. Graepel, Private traits and attributes are predictable from digital records of human behavior, *Proc. Natl. Acad. Sci. USA*, vol. 110, no. 15, pp. 5802–5805, 2013.

[79] L. Qi, W. Lin, X. Zhang, W. Dou, X. Xu, and J. Chen, A correlation graph based approach for personalized and compatible web APIs recommendation in mobile APP development, *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 5444–5457, 2023.

[80] S. Wu, S. Shen, X. Xu, Y. Chen, X. Zhou, D. Liu, X. Xue, and L. Qi, Popularity-aware and diverse web APIs recommendation based on correlation graph, *IEEE Trans. Comput. Soc. Syst.*, vol. 10, no. 2, pp. 771–782, 2023.

[81] X. Zhou, W. Liang, K. I. K. Wang, and L. T. Yang, Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations, *IEEE Trans. Comput. Soc. Syst.*, vol. 8, no. 1, pp. 171–178, 2021.

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

[83] E. Zeydan, E. Bastug, M. Bennis, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, Big data caching for networking: Moving from cloud to edge, *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 36–42, 2016.

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

[85] Z. Zheng, L. Song, Z. Han, G. Y. Li, and H. V. Poor, A stackelberg game approach to proactive caching in large-scale mobile edge networks, *IEEE Trans. Wirel.*

Commun., vol. 17, no. 8, pp. 5198–5211, 2018.
[86] A. Aijaz, M. Dohler, A. H. Aghvami, V. Friderikos, and M. Frodigh, Realizing the tactile Internet: Haptic communications over next generation 5G cellular networks, *IEEE Wirel. Commun.*, vol. 24, no. 2, pp. 82–89, 2017.
[87] X. Huang, R. Yu, J. Kang, Y. He, and Y. Zhang, Exploring mobile edge computing for 5G-enabled software defined vehicular networks, *IEEE Wirel. Commun.*, vol. 24, no. 6, pp. 55–63, 2017.
[88] E. Herrero, J. González, and R. Canal, Distributed cooperative caching, in *Proc. 2008 Int. Conf. Parallel Architectures and Compilation Techniques* (*PACT*), Toronto, Canada, 2017, pp. 134–143.
[89] S. Borst, V. Gupta, and A. Walid, Distributed caching algorithms for content distribution networks, in *Proc. 2010 IEEE INFOCOM*, San Diego, CA, USA, 2010, pp. 1–9.
[90] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, FemtoCaching: Wireless content delivery through distributed caching helpers, *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.
[91] J. Liu, B. Bai, J. Zhang, and K. B. Letaief, Content caching at the wireless network edge: A distributed algorithm via belief propagation, in *Proc. 2016 IEEE Int. Conf. Communications* (*ICC*), Kuala Lumpur, Malaysia, 2016, pp. 1–6.
[92] H. Tian, X. Xu, T. Lin, Y. Cheng, C. Qian, L. Ren, and M. Bilal, DIMA: Distributed cooperative microservice caching for Internet of Things in edge computing by deep reinforcement learning, *World Wide Web*, vol. 25, no. 5, pp. 1769–1792, 2022.
[93] W. Ali, S. M. Shamsuddin, and A. S. Ismail, A survey of web caching and prefetching, *Int. J. Adv. Soft Comput. Appl.*, vol. 3, no. 1, pp. 18–44, 2011.
[94] J. Ren, W. Qi, C. Westphal, J. Wang, K. Lu, S. Liu, and S. Wang, MAGIC: A distributed max-gain in-network caching strategy in information-centric networks, in *Proc. 2014 IEEE Conf. Computer Communications Workshops* (*INFOCOM WKSHPS*), Toronto, Canada, 2014, pp. 470–475.
[95] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, Web caching and Zipf-like distributions: Evidence and implications, in *Proc. IEEE INFOCOM '99. Conf. Computer Communications. Proceedings. Eighteenth Annual Joint Conf. IEEE Computer and Communications Societies. The Future is Now* (*Cat. No. 99CH36320*), New York, NY, USA, 1999, pp. 126–134.
[96] L. A. Adamic and B. A. Huberman, Zipf's law and the Internet, *Glottometrics*, vol. 3, no. 1, pp. 143–150, 2002.
[97] M. Cha, H. Kwak, P. Rodriguez, Y. Y. Ahn, and S. Moon, I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system, in *Proc. 7th ACM SIGCOMM Conf. Internet measurement*, San Diego, CA, USA, 2007, pp. 1–14.
[98] M. Ji, G. Caire, and A. F. Molisch, Fundamental limits of caching in wireless D2D networks, *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, 2016.
[99] M. Mahloo, P. Monti, J. Chen, and L. Wosinska, Cost modeling of backhaul for mobile networks, in *Proc. 2014 IEEE Int. Conf. Communications Workshops* (*ICC*), Sydney, Australia, 2014, pp. 397–402.
[100] S. Chia, M. Gasparroni, and P. Brick, The next challenge for cellular networks: Backhaul, *IEEE Microw. Mag.*, vol. 10, no. 5, pp. 54–66, 2009.
[101] S. Buzzi, I. Chih-Lin, T. E. Klein, H. V. Poor, C. Yang, and A. Zappone, A survey of energy-efficient techniques for 5G networks and challenges ahead, *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 697–709, 2016.
[102] X. Zhou, X. Yang, J. Ma, and K. I. K. Wang, Energy-efficient smart routing based on link correlation mining for wireless edge computing in IoT, *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14988–14997, 2022.
[103] J. Zhang, X. Zhang, M. A. Imran, B. Evans, and W. Wang, Energy efficiency analysis of heterogeneous cache-enabled 5G hyper cellular networks, in *Proc. 2016 IEEE Global Communications Conference* (*GLOBECOM*), Washington, DC, USA, 2016, pp. 1–6.
[104] S. Verdu and S. Shamai, Spectral efficiency of CDMA with random spreading, *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 622–640, 1999.
[105] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, Energy and spectral efficiency of very large multiuser MIMO systems, *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, 2013.
[106] C. Xiong, G. Y. Li, S. Zhang, Y. Chen, and S. Xu, Energy- and spectral-efficiency tradeoff in downlink OFDMA networks, *IEEE Trans. Wirel. Commun.*, vol. 10, no. 11, pp. 3874–3886, 2011.
[107] D. Wessels, *Web Caching*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2001.
[108] S. I. Ahmed, S. Y. Ameen, and S. R. M. Zeebaree, 5G mobile communication system performance improvement with caching: A review, in *Proc. 2021 Int. Conf. Modern Trends in Information and Communication Technology Industry* (MTICTI), Sana'a, Yemen, 2021, pp. 1–8.
[109] J. Cao, M. Ma, H. Li, R. Ma, Y. Sun, P. Yu, and L. Xiong, A survey on security aspects for 3GPP 5G networks, *IEEE Commun. Surv. Tutor.*, vol. 22, no. 1, pp. 170–195, 2020.
[110] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, 5G evolution: A view on 5G cellular technology beyond 3GPP release 15, *IEEE Access*, vol. 7, pp. 127639–127651, 2019.
[111] P. Gandotra, R. K. Jha, and S. Jain, A survey on device-to-device (D2D) communication: Architecture and security issues, *J. Netw. Comput. Appl.*, vol. 78, pp. 9–29, 2017.
[112] M. Waqas, Y. Niu, Y. Li, M. Ahmed, D. Jin, S. Chen, and Z. Han, A comprehensive survey on mobility-aware D2D communications: Principles. practice and challenges, *IEEE Commun. Surv. Tutor.*, vol. 22, no. 3, pp. 1863–1886, 2020.
[113] A. Liu and V. K. N. Lau, How much cache is needed to achieve linear capacity scaling in backhaul-limited dense

wireless networks? *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 179–188, 2017.
[114] P. B. Heidorn, Shedding light on the dark data in the long tail of science, *Libr. Trends*, vol. 57, no. 2, pp. 280–299, 2008.
[115] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration, *IEEE Commun. Surv. Tutor.*, vol. 19, no. 3, pp. 1657–1681, 2017.
[116] J. Pan and J. McElhannon, Future edge cloud and edge computing for Internet of Things applications, *IEEE Internet Things J.*, vol. 5, no. 1, pp. 439–449, 2018.
[117] D. W. Chadwick, W. Fan, G. Costantino, R. de Lemos, F. Di Cerbo, I. Herwono, M. Manea, P. Mori, A. Sajjad, and X. S. Wang, A cloud-edge based data security architecture for sharing and analysing cyber threat information, *Future Gener. Comput. Syst.*, vol. 102, pp. 710–722, 2020.
[118] X. Xu, H. Tian, X. Zhang, L. Qi, Q. He, and W. Dou, DisCOV: Distributed COVID-19 detection on X-ray images with edge-cloud collaboration, *IEEE Trans. Serv. Comput.*, vol. 15, no. 3, pp. 1206–1219, 2022.
[119] P. Cao, E. W. Felten, A. R. Karlin, and K. Li, A study of integrated prefetching and caching strategies, *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 23, no. 1, pp. 188–197, 1995.
[120] S. Ioannidis and E. Yeh, Adaptive caching networks with optimality guarantees, *IEEE/ACM Trans. Netw.*, vol. 26, no. 2, pp. 737–750, 2018.
[121] P. Yang, N. Xiong, and J. Ren, Data security and privacy protection for cloud storage: A survey, *IEEE Access*, vol. 8, pp. 131723–131740, 2020.
[122] J. Ni, K. Zhang, and A. V. Vasilakos, Security and privacy for mobile edge caching: Challenges and solutions, *IEEE Wirel. Commun.*, vol. 28, no. 3, pp. 77–83, 2021.
[123] L. Xiao, X. Wan, C. Dai, X. Du, X. Chen, and M. Guizani, Security in mobile edge caching with reinforcement learning, *IEEE Wirel. Commun.*, vol. 25, no. 3, pp. 116–122, 2018.
[124] Z. Li, X. Xu, X. Cao, W. Liu, Y. Zhang, D. Chen, and H. Dai, Integrated CNN and federated learning for COVID-19 detection on chest X-ray images, *IEEE/ACM Trans. Comput. Biol. Bioinform.*, doi: 10.1109/TCBB.2022.3184319.
[125] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, Mobility-aware proactive edge caching for connected vehicles using federated learning, *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5341–5351, 2021.
[126] S. Liu, C. Zheng, Y. Huang, and T. Q. S. Quek, Distributed reinforcement learning for privacy-preserving dynamic edge caching, *IEEE J. Sel. Areas Commun.*, vol. 40, no. 3, pp. 749–760, 2022.
[127] Z. Ning, K. Zhang, X. Wang, L. Guo, X. Hu, J. Huang, B. Hu, and R. Y. K. Kwok, Intelligent edge computing in Internet of vehicles: A joint computation offloading and caching solution, *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2212–2225, 2021.
[128] Z. Li, M. A. Uusitalo, H. Shariatmadari, and B. Singh, 5G URLLC: Design challenges and system concepts, in *Proc. 2018 15th Int. Symp. on Wireless Communication Systems* (*ISWCS*), Lisbon, Portugal, 2018, pp. 1–6.
[129] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, Deep-learning-enhanced multitarget detection for end–edge–cloud surveillance in smart IoT, *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12588–12596, 2021.
[130] W. Liang, Y. Hu, X. Zhou, Y. Pan, and K. I. K. Wang, Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial IoT, *IEEE Trans. Ind. Inform.*, vol. 18, no. 8, pp. 5087–5095, 2022.
[131] I. Zyrianoff, A. Trotta, L. Sciullo, F. Montori, and M. Di Felice, IoT edge caching: Taxonomy, use cases and perspectives, *IEEE Internet Things Mag.*, vol. 5, no. 3, pp. 12–18, 2022.
[132] X. Xu, J. Gu, H. Yan, W. Liu, L. Qi, and X. Zhou, Reputation-aware supplier assessment for blockchain-enabled supply chain in industry 4.0, *IEEE Trans. Ind. Inform.*, vol. 19, no. 4, pp. 5485–5494, 2023.
[133] L. Qi, Y. Yang, X. Zhou, W. Rafique, and J. Ma, Fast anomaly identification based on multiaspect data streams for intelligent intrusion detection toward secure industry 4.0, *IEEE Trans. Ind. Inform.*, vol. 18, no. 9, pp. 6503–6511, 2022.
[134] S. Gu, Y. Wang, N. Wang, and W. Wu, Intelligent optimization of availability and communication cost in satellite-UAV mobile edge caching system with fault-tolerant codes, *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 4, pp. 1230–1241, 2020.

Hanwen Li is currently pursuing the BS degree in computer science and technology at School of Computer Science, Nanjing University of Information Science and Technology, China. His main research interests include edge computing and deep learning.

Mingtao Sun received the BS degree from Chaohu University, China in 2017, majoring in computer science. He is currently a lecturer at Weifang University of Science and Technology, China. His main research interests include computer theory and animation production.

Fan Xia is currently pursuing the BS degree in data science at Reading Academy, Nanjing University of Information Science and Technology, China. Her main research interests include edge computing, knowledge graph, and deep learning.

Muhammad Bilal received the PhD degree in information and communication network engineering from Korea University of Science and Technology, Republic of Korea in 2017. From 2017 to 2018, he was at Korea University, Republic of Korea, where he was a post-doctoral research fellow at Smart Quantum Communication Center. In 2018, he joined the Hankuk University of Foreign Studies, Republic of Korea, where he was an associate professor. He is currently a senior lecturer at School of Computing and Communications, Lancaster University, UK. He is the author/coauthor of more than 100 articles published in renowned journals, one book editorship, three issued US patents, and six Korean patents. His research interests include network optimization, cyber security, the Internet of Things, vehicular networks, information-centric networking, digital twin, artificial intelligence, and cloud/fog computing. He has served as a technical program committee member on many international conferences, including the IEEE VTC, the IEEE ICC, ACM SigCom, and the IEEE CCNC. He serves as an editor for the *IEEE Transactions on Intelligent Transportation Systems*, *Alexandria Engineering Journal* (Elsevier), *Physical Communication* (Elsevier), *Computer Systems Science and Engineering*, *Intelligent Automation & Soft Computing*, *IEEE Future Directions in Technology*, *Policy*, *and Ethics Newsletter*, *Frontiers in Communications and Networks*, and *Frontiers in the Internet of Things*, and the co-editor-in-chief of the *International Journal of Smart Vehicles and Smart Transportation*.

Xiaolong Xu received the PhD degree from Nanjing University, China in 2016. From 2017 to 2018, he worked as a researcher at the Department of Computer Science and Engineering, Michigan State University, USA. He is currently a professor at School of Software, Nanjing University of Information Science and Technology, China. His research interests include cloud computing, big data, edge computing, deep learning, and federated edge learning. He has published over 100 peer-review papers in international journals and conferences, including *IEEE TPDS*, *IEEE TKDE*, *IEEE TSC*, *ACM TOSN*, *IEEE TITS*, *IEEE TII*, *ACM TOIT*, *ACM TOMM*, *ACM TIST*, *IEEE TVT*, *IEEE IOT*, *IEEE TCC*, *IEEE TBD*, *IEEE TCSS*, *IEEE TETCI*, *Software*: *Practice and Experience*, *World Wide Web* journal, *Information Sciences*, *Journal of Network and Computer Applications*, *Future Generation Computer Systems*, *Computational Intelligence*, *Concurrency and Computation*: *Practice and Experience*, *IEEE ICWS*, *ICSOC*, etc. He was selected as the Highly Cited Researcher of Clarivate 2021 and 2022. He received the Best Paper Award from the *journal of Network and Computer Applications* at 2022, the Editor's Choice Paper Award from *Future Generation Computer Systems* at 2022, the Top Citation Award from *Computational Intelligence* journal, the Best Paper Awards from IEEE CBD 2016, IEEE CPSCom 2020, SPDE 2020, IEEE CyberSciTech 2021, IEEE ATC 2022, IEEE iThings 2022, CENET 2021, EAI Cloudcomp 2021, the Outstanding Paper Award from IEEE SmartCity 2022, EAI Cloudcomp 2019, the Best Student Paper Award from EAI Cloudcomp 2019, and the Best Session Paper from IEEE DSAA 2020. He also received the Outstanding Leadership Award from IEEE UIC 2022.