IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 22, 2020, accepted November 3, 2020, date of publication November 11, 2020,
date of current version November 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3037357

A Classification of the Enabling Techniques for Low Latency and Reliable Communications in 5G and Beyond: AI-Enabled Edge Caching

LILIAN CHARLES MUTALEMWA AND SEOKJOO SHIN, (Member, IEEE)
Department of Computer Engineering, Chosun University, Gwangju 61452, South Korea
Corresponding author: Seokjoo Shin (sjshin@chosun.ac.kr)

This work was supported by the Chosun University through the Research Fund.

ABSTRACT Various advanced and mission-critical applications are enabled by the emerging technologies in fifth-generation (5G) mobile communication systems. To ensure improved quality of experience (QoE) of users, 5G and beyond networks require ultra-reliable low-latency communications (URLLC). The successful realization of the URLLC entails the advent of new technological concepts. Therefore, this article presents an overview of the enabling techniques for the URLLC. Classification of the enabling techniques is done and an extensive review of the literature is presented to identify the state-of-the-art techniques, limitations, and the potential approaches for alleviating the limitations. It is observed that artificial intelligence (AI)-enabled edge computing and caching solutions are widely explored as promising techniques to effectively guarantee low latency and reliable content acquisition while reducing redundant network traffic and improving the QoE. Therefore, we present a classification of the AI-enabled edge caching solutions and discuss various mechanisms of the caching agents. In particular, we investigate the use of deep learning (DL), deep reinforcement learning (DRL), and federated learning (FL) algorithms. Subsequently, we analyze the performance of the state-of-the-art edge caching schemes and demonstrate the performance gains of FL frameworks over conventional centralized and decentralized DL and DRL frameworks. We confirm that FL edge caching is a viable mechanism in 5G and beyond networks. On the other hand, it is shown that the IEEE 802.1 time sensitive networking and the emerging IETF deterministic networking standards present effective mechanisms when deterministic networks with bounded ultra-low latency are considered. Finally, we present the open issues and opportunities for further research.

INDEX TERMS 5G, ultra-reliable low-latency communications, edge computing, caching, federated learning, time sensitive network, deterministic network.

I. INTRODUCTION

Emerging technologies and new applications such as intelligent transport systems, industrial automation, remote patient diagnosis and surgery, smart homes, and serious gaming require ultra-reliable and low-latency communications. The requirements cannot be fulfilled by the existing fourth-generation (4G) mobile communication networks. Therefore, fifth-generation (5G) mobile communication has been devised. The 5G networks present some extensive improvements in the 3rd generation partnership project (3GPP's) long term evolution (LTE) technology. The main goals of the 5G networks include improved capacity, reliability, and energy efficiency while reducing latency and significantly increasing connection density [1], [2]. Comparing the requirements of 5G with the 4G systems, the 5G systems must ensure a 10× decrease in latency, a 10× improvement in throughout, a 100× increase in the traffic capacity, and a 100× improvement in the network efficiency [3].

According to the international telecommunication union radiocommunication sector, 5G wireless systems are designed to support three generic services. The services are classified as enhanced mobile broadband (eMBB), ultra-reliable low-latency communications (URLLC), and massive machine-type communications (mMTC) [2], [4]–[7]. The eMBB supports stable connections with very high peak

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

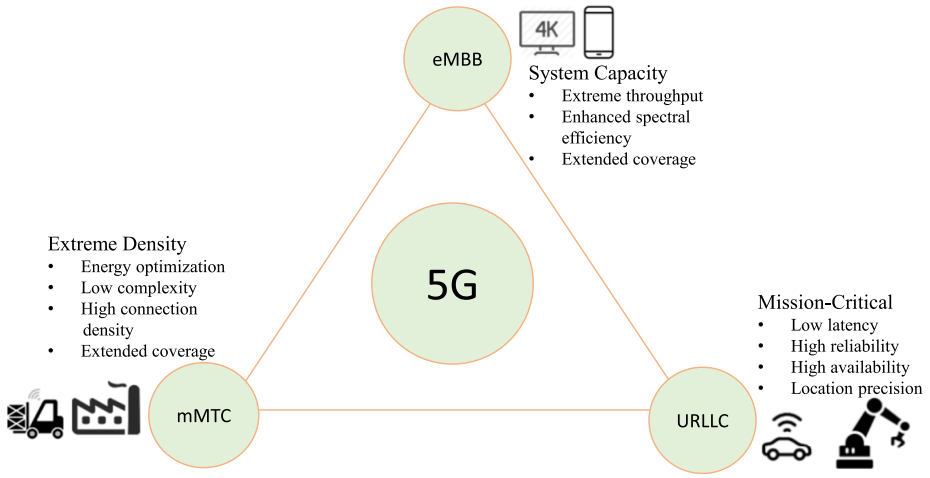

FIGURE 1. Features and services of 5G [2], [4].

data rates, as well as moderate rates for cell-edge users. The mMTC provides efficient wireless connectivity for massive number of Internet of things (IoT) devices, which are only periodically active and send small data payloads. The URLLC provides critical communication services to support mission-critical applications requiring low-latency transmissions of small payloads with very high reliability. The features of the 5G services are presented in Fig. 1.

Beyond 5G (B5G) networks are motivated by the machine-based vertical applications such as autonomous driving, unmanned aerial vehicle services, extended reality, autonomous services, and industrial automation [8]–[10]. Thus, as compared to the URLLC in 5G cellular networks, many B5G networks will have stricter requirements in terms of latency and reliability. Furthermore, the B5G networks will demand absolute time-synchronization and on-time delivery of packets for deterministic and isochronous real-time applications [11]. Therefore, achieving ultra-low latency (ULL) is crucial for 5G and B5G networks.

In the case of URLLC, traffic flows require extremely low delays on the order of 1 ms, with very high reliability of about 99.999% or 1-10^{-5} block error rate (BLER) for short packets of up to 32 bytes [1], [3], [12]–[15]. Furthermore, some applications require ULL below 1 ms [13], [16]. For example, the end-to-end latencies for industrial automation should be on the order of a few μs to a few ms, around 1 ms or below for the Tactile Internet (TI), and on the order of 100 μs for the one-way fronthaul in wireless cellular networks. Also, near real-time connectivity is required in robotic surgery and intelligent transport systems. Moreover, both high data rates and ULL are necessary in applications such as autonomous automotive vehicles, augmented and virtual reality, and robotic applications that are essential for industrial IoT. The high data rates may be required for transporting video feeds from cameras that are used to control the vehicles and robots [16]. Therefore, achieving URLLC and deterministic ULL will enable the successful realization of the 5G and B5G applications [4], [12], [17]–[19]. Table 1 presents some advantages of URLLC and ULL in the 5G and B5G applications, according to [1], [4], [20].

A. MOTIVATION

This study is motivated by the findings and discussions of the recent studies in [9], [16], [21]–[32]. According to [21], reliability and latency guaranteeing schemes for 5G services often consider three aspects. (1) Diversity in the space, frequency, time, and code domains can guarantee reliability in 5G. Thus, multiuser detection schemes and accurate channel estimation can improve reliability. (2) Latency can be reduced by reshaping the frame structure. When the system operates above 6 GHz, subcarrier spacing of orthogonal frequency division multiplexing (OFDM) symbols can be increased from 15 to 60, 120, or 240 kHz to reduce the slot duration from 1 ms to 10 μs. Hence, the transmission time interval can be shortened proportionally to the duration of one slot to reduce latency. Furthermore, full duplex (FD) schemes can be employed to operate the uplink and downlink simultaneously. FD supports instant feedback, such as early acknowledgment (ACK) and negative ACK (NACK), which result in reduced retransmission latency. (3) Schemes such as non-orthogonal multiple access (NOMA) and orthogonal multiple access (OMA), which allow reuse of radio resources, can enhance both reliability and latency. Also, in NOMA, more than one user can be served within the same radio resource to improve the spectrum efficiency.

In [22], it was deliberated that techniques such as dynamic multiplexing, latency sensitive scheduling schemes, and grant-free access can be effective at reducing latency in 5G. The work in [23], [29] promoted the use of network slicing technique to effectively allocate network resources to provide performance guarantees for URLLC services. In [9], [24], [25], [27], [28], [30] it was discussed that 5G networks would experience a shift from the conventional cloud computing setting to edge computing systems.

TABLE 1. Advantages of ultra-reliable and low-latency communications in mission-critical applications.

Use case	Application	Requirements	Advantages
Healthcare	Remote robotic surgery and patient diagnosis.	Latency on the order of 1 ms with data rate of 100 Mbps.	Ensures signals are transmitted with minimum latency and very high reliability to minimize harm to the patients. Also, robotic surgery is beneficial because under controlled conditions, robots can be efficient and tireless.
Intelligent transport systems	Remote driving, self-driven cars, and traffic management.	Latency on the order of 10-100 ms with data rate on the order of 700 Mbps for road safety related cases. Latency on the order of 1 ms for remote driving.	Ensures reliable signal transmission for real-time services to enable quick responses to changing road conditions and effective cooperative vehicle maneuvering. Thus, minimize the probability of errors and accidents.
Industrial automation and smart grid	Automated assembly lines, control systems, and power grid management.	Latency on the order of 0.25 ms for machine tools operation. Latency in the order of 1 ms for dynamic activation and deactivation in smart grid.	Ensures signal transmission with minimum latency and very high reliability to minimize errors and damage. Automation allows companies to improve their productivity and efficiency of manufacturing and industrial processes.
Serious gaming and entertainment	Online gaming and cloud-based entertainment.	Latency on the order of 1 ms with data rate of 1Gbps in the case of serious gaming.	Ensures minimum latency to avoid jitter in the video and audio. Thus, avoid frustrations to the gamers and ensure users experience real-time, interactive, and immersive entertainment.
Education and culture	Remote learning/education.	Round trip latency of 5-10 ms with data rates as high as 1 Gbps.	Facilitates multi-modal human-machine interfaces for perceivable visual, auditory, and haptic interaction. Also, enables remote play of musical instruments.

Edge computing systems deploy computational power to the network edges to meet the requirements of low latency, high reliability, as well as supporting resource- constrained nodes which are reachable only over unreliable network connections. Moreover, future networks will be able to utilize local data to conduct intelligent inference and control. Therefore, it was presented in [24] that a new machine learning model has emerged, namely federated learning (FL). FL provides smart models, low latency, reduced power consumption, and allows the use of training data from a user equipment (UE) without sacrificing the personal data privacy. In [26], it was demonstrated that in-network caching is an effective technique for reducing content delivery latency and to improve content availability in IoT systems.

The work in [16] presented that the ULL communications in 5G and beyond networks will likely rely on the IEEE 802.1 time sensitive networking (TSN) and the emerging IETF deterministic networking (DetNet) standards. In [31]–[33], it was highlighted that the TSN and DetNet standards could establish reliable communication channels for a given traffic flow over the network with deterministic quality of service (QoS) guarantees. The standards can guarantee QoS in terms of bounded ULL, jitter, congestion loss, and reliability, regardless of the other ongoing flows competing for the same resources. For example, in the use case of industrial automation, TSN can provide time synchronization and tight gating mechanisms to allow strict and time-bounded data delivery of time-sensitive traffic flows such as sensor data, control input to actuators, and audio or video packets [11].

Therefore, this study reviews the literature and presents an overview of the techniques which utilize the approaches presented in [9], [16], [21]–[32]. Specifically, we explore the mechanisms of configurable subcarrier spacing, grant-free access, latency sensitive scheduling schemes, dynamic multiplexing schemes, network slicing, edge computing, edge caching, in-network caching, TSN, and DetNet. Thereafter, we discuss the performance features of various edge caching frameworks based on deep learning (DL), deep reinforcement learning (DRL), and FL algorithms. Also, we conduct a comparative analysis of the state-of-the-art centralized and decentralized edge caching frameworks.

B. LITERATURE REVIEW

Recently, many studies are focusing on reviewing the enabling techniques for low latency and reliable communications in 5G and beyond networks. The existing studies focused on surveying various important techniques and mechanisms. However, many of the studies are limited to specific approaches and potential use cases. For example, although [4] aimed at reviewing the techniques for URLLC in accordance with the approaching era of technology and industry requirements while highlighting a few implementation issues of URLLC, the study did not discuss the mechanisms of edge caching or in-network caching. In [3], a holistic view on wireless TI was presented along with a thorough review of the existing state-of-the-art to identify and analyze the involved technical issues and highlight potential solutions. It also focused on various issues including the main technical requirements, the key application areas, security, and privacy issues of TI applications. However, [3] presented a brief overview of the enabling techniques. It did not discuss the state-of-the-art mechanisms for several enabling techniques. For instance, the study in [3] only mentioned about TSN but failed to give details about the TSN standards.

Also, [3] highlighted the need for robust and deterministic wireless connectivity but it did not explore the mechanisms of the DetNet standard. Moreover, [3] provided a general overview of various techniques, but it did not discuss the existing edge caching algorithms. Similarly, the study in [9] provided a holistic overview of edge computing techniques and its potential use cases and applications in 5G and beyond. However, [9] did not provide detailed discussions about the state-of-the-art DL, DRL, and FL edge caching algorithms. Moreover, [9] did not include performance analysis of the caching schemes.

In [28], the techniques for achieving low latency in applications such as intelligent Internet of vehicles and autonomous driving were discussed. However, [28] did not discuss the techniques for achieving bounded ULL. For example, [28] did not include the mechanisms of TSN and DetNet standards in its discussions. In [1], various techniques for low latency in 5G were discussed. However, [1] only mentioned about various caching strategies. The study did not explore the state-of the-art AI-enabled caching strategies. Furthermore, the work in [34] presented a review of the key issues in mobile edge caching and discussed the existing learning-based caching solutions. However, [34] failed to discuss the mechanisms of FL edge caching frameworks. On the other hand, [35], [36] limited their discussions to in-network caching strategies for the IoT networks while [37], [38] focused mainly on the recent advances of edge caching techniques. Moreover, [28], [38] pointed out that it is important to optimize the edge caching while guaranteeing the privacy of personal data. However, [28], [38] failed to discuss the privacy preserving mechanisms of the FL edge caching frameworks. Various issues of FL schemes and FL use case scenarios were considered in [39], [40] while [41] discussed the state-of-the-art strategies of machine learning techniques for edge caching. However, [39], [41], [42] failed to present performance analysis of the state-of-the-art edge caching schemes. On the other hand, [43] compared the performance of the FL and centralized schemes. However, the analysis was limited to the privacy performance of the schemes. Thus, [43] did not analyze the latency performance of the schemes.

On the other hand, it is important to achieve bounded ULL in applications such as autonomous automotive vehicles and industrial automation and control systems. Nevertheless, the studies in [1], [3], [4], [9], [28], [34]–[39], [41], [43] failed to discuss the enabling techniques for achieving bounded ULL. In contrast, bounded ULL was considered in [16], [31], [44]. Therefore, [16], [31], [44] discussed the techniques for achieving bounded ULL in deterministic networks. The work in [16] surveyed the existing studies that specifically target the support of ULL in 5G networks. Nonetheless, [16] limited its scope and did not discuss in detail the other URLLC enabling techniques such as caching. In [31], the applicability of TSN and DetNet mechanisms to various industries such as industrial automation and automotive in-vehicle networking was discussed. On the contrary, [31], [44] did not discuss the other URLLC enabling techniques such as caching, network slicing, sTTI, and grant-free access.

Therefore, this survey can be distinguished from the existing surveys. The main difference is based on the fact that this study discusses a wide range of techniques. In particular, this study presents detailed discussions about numerous mechanisms for grant-free access, latency sensitive scheduling, dynamic multiplexing, network slicing, edge computing, edge caching, sTTI, in-network caching, TSN, and DetNet. Therefore, we consider not only the state-of-the-art techniques for URLLC, but also the emerging approaches for achieving deterministic networking and bounded ULL. Furthermore, in the case of edge caching techniques, we analyze the performance of various caching algorithms, including the state-of-the-art DL, DRL, and FL algorithms.

C. CONTRIBUTION AND PAPER ORGANIZATION

The main contributions of this study can be summarized as follows. (1) Provide an overview and a classification of the enabling techniques for URLLC. (2) Discuss the techniques for achieving URLLC and deterministic ULL while highlighting the viable mechanisms for various mission-critical applications, limitations of the mechanisms, and potential solutions for the limitations. (3) Enlist and discuss the key performance features of the state-of-the-art AI-enabled edge caching frameworks focusing on the caching agent strategies, learning algorithms, network environments, and the potential use cases. (4) Conduct a classification and performance analysis of the AI-enabled edge caching schemes to exhibit the performance gains of FL frameworks over DL and DRL frameworks. (5) Outline the technical challenges and open issues for further research.

The remainder of this article is organized as follows. Section II presents a review of the literature on the enabling techniques for URLLC and deterministic ULL. Section III discusses the state-of-the-art caching solutions to ensure low content delivery latency and improved content acquisition reliability in IoT networks. Section IV summarizes the limitations of the enabling techniques and highlights the methods for mitigating the limitations. Section V presents a classification and performance analysis of the AI-enabled edge caching solutions. It includes investigations on the performance of DL, DRL, and FL edge caching mechanisms. Section VI provides a summary and discussions. The open issues and opportunities for further research are discussed in section VII. In section VIII, the paper is concluded.

II. ENABLING TECHNIQUES FOR LOW LATENCY AND RELIABLE COMMUNICATIONS IN 5G AND B5G

Ensuring URLLC in mission-critical applications is of capital importance [45]. However, it is a challenging task to achieve high reliability and low latency simultaneously. In this study, we classify the key enabling techniques into four categories based on their main objectives, as shown in Fig. 2. The techniques for low latency include the use of configurable subcarrier spacing and short transmission time interval (sTTI),

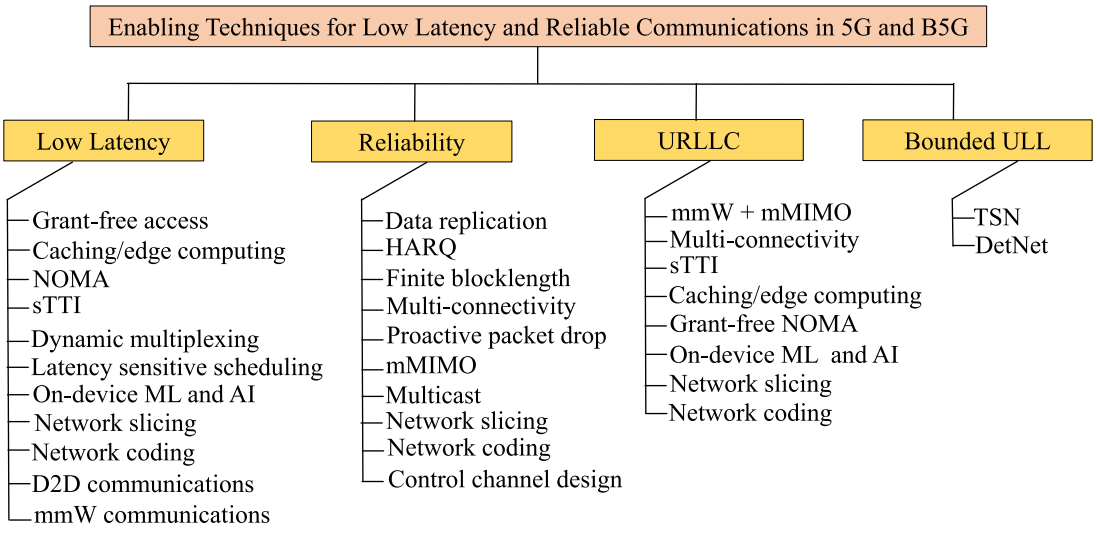

FIGURE 2. Classification of the enabling techniques for low latency and reliable communications.

grant-free access, non-orthogonal multiple access (NOMA), edge computing, caching, dynamic multiplexing, latency sensitive scheduling schemes, network coding, network slicing, on-device machine learning (ML), and AI [2], [7], [19], [21], [42], [45]–[48]. Also, low latency can be achieved through millimeter wave (mmW) communications and device-to-device (D2D) communications [1], [15], [42], [47]. The techniques for high reliability include network coding, network slicing, spatial diversity with massive multiple input multiple outputs (mMIMO) antennas, hybrid automatic repeat request (HARQ) with sTTI, finite blocklength, and multi-connectivity [2], [7], [19], [21], [42], [45]–[48]. Also, useful mechanisms such as multicasting, data replication, control channel design, and proactive packet drop can guarantee high reliability [45]. Reliability can also be achieved through the use of unlicensed and licensed spectrum coordination mechanisms [49], [50]. The techniques for URLLC include multi-connectivity, network coding, mMIMO with mmW communications, sTTI, network slicing, grant-free access and NOMA, edge computing, and caching [1], [2], [7], [15], [19], [21], [42], [45]–[48]. Some of the techniques, such as edge caching, scheduling schemes, and network slicing can be enabled through the use of ML algorithms and AI.

When bounded ULL is required, i.e., deterministic latency where all frames of a given application traffic flow must not exceed a prescribed ULL bound, the TSN and DetNet standards present effective mechanisms [10], [16], [31]–[33], [51]. Thus, the TSN and DetNet standards ensure real-time guarantees, determinism, high fault-tolerance, and clock synchronization mechanisms. The techniques are discussed below. Thereafter, we present detailed discussions and performance investigations of various edge caching algorithms.

A. CONFIGURABLE SUBCARRIER SPACING AND sTTI

While LTE uses fixed numerology of 15 kHz subcarrier spacing, it is possible to use configurable subcarrier spacing in 5G with values of 15, 30, and 60 kHz for below 6 GHz, and 120 and 240 kHz for above 6 GHz band [15], [19]. The configurable subcarrier spacing accommodates a different number of slots within a 1 ms subframe and obtains the transmission time interval (TTI) of 1, 0.5, 0.25, 0.125, and 0.0625 ms, respectively. However, the highest subcarrier spacing that supports data transmissions is 20 kHz, corresponding to a TTI of 0.125 ms. Therefore, the different numerologies lead to different slot lengths, ranging from 1 ms at 15 kHz subcarrier spacing to 0.125 ms at 120 kHz subcarrier spacing, enabling shorter TTIs.

Thus, 5G uses sTTI and short frame structure to ensure reduced transmission latency [2], [15], [21], [52]. Transmission latency smaller than 1 ms is achieved by controlling the symbol period and the number of symbols in a packet [6]. The use of sTTI is also enabled by using fewer OFDM symbols per TTI and shortening the OFDM symbols through wider subcarrier spacing [45]. Although the use of sTTI is useful in reducing latency, it can result in some negative effects. For example, sTTI can introduce more control overhead which results in inefficient resource utilization [19], [45]. To address the limitation of increased control overhead, grant-free access is used in the uplink transmissions [17], [45].

B. GRANT-FREE ACCESS

When grant-free access is used, it ensures dynamic uplink scheduling and shortens the procedure for uplink resource assignment by skipping the resource reservation phase [19], [45], [53]. As a result, it reduces the latency, control overhead, and energy consumption, especially at the UE with longer sleeping time [54], [55]. For example, with grant-free access, a UE can send data along with the preamble used for establishing a link [15]. Grant-free access is more useful in the uplink communications because, in the uplink, the UE sends a scheduling request to the serving base station in a dedicated and periodic resource and then

waits for the scheduling grant from the base station in some slots later. The process can introduce delay. On the other hand, for downlink communication, downlink scheduling and data transmission can happen in the same transmission interval. Hence, it is less complex to address the challenges in downlink communications [56]. The drawback of grant-free access is that it can increase the number of collision events. Consequently, the reliability may be reduced [15], [53], [57]. It was presented in [15] that the delivery reliability in grant-free access can be improved by sending replicas of the same packet. However, it was demonstrated in [53] that grant-free scheduling solutions might not be able to guarantee the required URLLC under the presence of sporadic traffic. In particular, the method of transmitting a few replicas per packet might be inadequate to support sporadic traffic which requires very low latency. This is mainly due to the impact of self-collisions.

The design and feasibility of grant-free access for uplink URLLC were also investigated in [56]. It was shown that contention based grant-free transmission with resources shared by multiple users may be more efficient than the grant-based or traditional semi-persistent scheduling transmission, and it can well meet the reliability requirement of URLLC within the latency bound. Furthermore, it was shown that the data collision could be handled by the use of HARQ design with acknowledgment for early termination. Also, frequency hopping along with retransmission slots can further improve the channel diversity and enhance the reliability. Moreover, it was shown that NOMA schemes such as sparse code multiple access (SCMA) or advanced receivers such as expectation propagation algorithm or message passing algorithm could improve the reliability, reduce latency, and improve resource efficiency. A grant-free NOMA with an advanced receiver scheme was also presented in [57].

The effectiveness of various NOMA-enabled grant-free schemes was shown in [54]. The schemes were proposed by the 3GPP with the main aim of supporting uplink transmissions for massive connectivity and grant-free transmission procedures with low latency and high reliability. It was shown that the NOMA-enabled schemes present competitive solutions for small packet transmissions in many application scenarios. In [21], it was shown that SCMA and full duplex schemes could be combined to improve the BLER performance by exploiting diversity gains in the space, frequency, and code domains. In [58]–[63], the latency in SCMA was reduced through the use of low complexity multiuser detection schemes.

C. DYNAMIC MULTIPLEXING

From a system design perspective, it is necessary to allow URLLC and other types of traffic such as eMBB to coexist in the same network resources [7], [22], [29], [64]. Therefore, dynamic multiplexing is required in hybrid services, such as when eMBB and URLLC services share the same radio spectrum. Dynamic multiplexing enables the coexistence of the eMBB and URLLC services to best accommodate the mixed demands in a cost-effective way. Furthermore, dynamic multiplexing ensures maximized spectral efficiency of both services [17], [22]. Also, due to the different scheduling granularity, eMBB and URLLC services need to be dynamically multiplexed in the time domain. This is because URLLC traffic is characterized by small and sporadic packets which require low latency and high reliability while eMBB services are featured with large payloads, high peak data rates, or other high bandwidth, to guarantee sufficiently high and stable quality of data such as image or voice contents [23].

Therefore, to ensure low latency, when the URLLC packets arrive during an ongoing eMBB transmission, the packets will not wait until the end of the eMBB burst to be scheduled. The URLLC traffic is immediately scheduled upon arrival in the next mini-slot [13]. The base station may signal to suspend the eMBB transmission to free up resources to transmit the URLLC data in the next URLLC TTI. The process is referred to as puncturing the current eMBB transmission or preemption indication [2], [17], [22], [45], [52]. When the puncturing process occurs, an indication mechanism is useful for the base station to inform the punctured eMBB user about the location of the preempted resources to improve its decoding performance.

Preemption techniques may be implemented in the uplink or downlink [22]. However, preemption in the uplink is more challenging to implement than in the downlink [17], [22]. In the uplink, the preemption is performed by the user, while in the downlink, it is performed by the base station. As a result, the uplink preemption indication must be signaled to the eMBB user prior to the URLLC uplink transmission to avoid causing interference to the transmission. For the downlink case, the preemption indication is transmitted after the actual downlink data transmission [22]. A limitation of the uplink preemption indication is that it imposes higher requirements on the eMBB UE processing capability. To address the limitation, an enhanced power control mechanism is employed at the URLLC UE [22].

In the enhanced power control mechanism, the base station can schedule an urgent URLLC transmission on top of an ongoing eMBB transmission, and indicate the URLLC UE to boost its power to guarantee the reliability [13], [22]. Therefore, the eMBB and URLLC transmissions are superpositioned on the overlapping resources through the use of NOMA. A significant limitation of the enhanced power control mechanism is that it is not useful if the URLLC UE is already power-limited. For example, the mechanism is not helpful if the UE is located at the cell edge [22]. Furthermore, it shows that the power control mechanism may not be appropriate to support the dynamic superposition of eMBB and URLLC transmissions. This is mainly because the URLLC user is often required to boost its power by a significant amount when its transmission is overlapping the eMBB transmission. Two methods which can alleviate the limitations of the enhanced power control mechanism are enhanced open-loop and enhanced closed-loop power control mechanisms [22]. The work in [22] pointed out that uplink

preemption indication may also be very beneficial for grant-free uplink transmissions. The work in [13], [65] presented some of the state-of-the-art algorithms for joint scheduling of URLLC and eMBB traffic in the emerging 5G systems.

D. LATENCY SENSITIVE SCHEDULING SCHEMES

Data packets in URLLC services are generated abruptly. To ensure low latency, two scheduling schemes are adopted: instant scheduling or reservation-based scheduling [6]. In the instant scheduling scheme, any ongoing data transmission is interrupted to initiate the URLLC packet. The scheme is effective at reducing the URLLC access time but it causes severe performance degradation. In the reservation-based scheduling scheme, URLLC resources are reserved prior to the data scheduling through semi-static or dynamic reservations [6]. In the semi-static reservation scheme, the base station infrequently broadcasts the configuration of the frame structure such as frequency numerology and service period. Semi-static reservation may guarantee high reliability and low latency. However, it results in inefficient resource utilization. Up to half of the URLLC resources may be wasted under the semi-static resource partition due to queuing effect and low trunking efficiency [22].

In the dynamic reservation scheme, information on the URLLC resource is updated frequently using the control channel of a scheduled user. For example, if an eMBB packet consists of 14 symbols, 10 symbols are used for the eMBB transmission, and the rest is reserved for the URLLC service. A limitation of the dynamic reservation scheme is that, when there is no URLLC transmission in the scheduled period, resources reserved for the URLLC service are wasted. Furthermore, the dynamic reservation scheme requires additional control overhead to indicate reservation information [6]. It was shown in [6] that the instant scheduling strategy could outperform reservation-based scheduling in terms of the average latency but it causes a throughput loss of the eMBB service. Also, the dynamic reservation can outperform the semi-static reservation in terms of the latency due to the fast resource adaptation.

E. NETWORK SLICING

Network slicing refers to the process of slicing a physical network into logical subnetworks optimized for specific applications and services. It aims to allocate dedicated network resources for verticals of interest [23], [29], [45], [64], [66]. Network slicing is an effective technique for meeting the diverse deployment requirements of URLLC, eMBB, and mMTC services in a multi-tenant 5G network. It allows service coexistence and heterogeneity through which each service is allocated network computing and communication resources to provide performance guarantees and isolation from the other services [5], [29], [64], [66]. Mission-critical applications such as vehicle-to-vehicle communication and industrial automation can benefit from network slicing. In these applications, it is difficult to model and predict queuing delays with very high accuracy. For example, in the industrial automation scenario, owing to its diverse set of application requirements, a network slice can be allocated for a high-definition video transmission between a person remotely monitoring a process (or robot) and another slice to provide an ultra-reliable transmission between sensors, controllers and actuators [45]. Several network slicing algorithms were presented in [23], [29], [64], [66], [67].

Network slicing for eMBB and URLLC uplink communication can be achieved through orthogonal slicing or non-orthogonal slicing [5], [68]. Heterogeneous orthogonal multiple access (H-OMA) schemes may be used in the orthogonal slicing while heterogeneous non-orthogonal multiple access (H-NOMA) schemes are used in the non-orthogonal slicing [5]. In NOMA, multiple users are scheduled non-orthogonally on the same spectrum resource [46]. The users send data simultaneously over the non-orthogonal channels. The signals create inter-user interference, but the base station can employ successive interference cancellation (SIC), i.e., decode one of the signals and then subtract the corresponding codeword from the received signal, such that the other signal is interference-free. As a result, NOMA can increase spectral efficiency and improve reliability [14], [55], [68]–[70].

The work in [5] revealed that in H-NOMA the URLLC device is decoded first. This is mainly because URLLC decoding cannot depend on the decoding of eMBB whose reliability and latency requirements are more relaxed. It also showed that H-NOMA can be more advantageous than H-OMA depending on the application scenarios. Also, H-NOMA is always beneficial when the eMBB rate is very large. The work in [46] investigated the advantages of relaying in NOMA and proposed an optimal design on relaying-enabled URLLC networks. Some interesting design issues for low-latency NOMA systems were also considered in [45], [71].

The work in [14] considered the issue of energy-efficient transmissions for URLLC multiuser channels. The work pointed out that the SIC technique may not always be feasible since there exist situations where none of the receivers can perform SIC and decode messages of the other users. Therefore, it presented a superposition coding techniques for energy-efficient resource allocation for a two-user heterogeneous NOMA downlink with a finite blocklength code. However, it was observed that the superposition coding-based NOMA is more energy efficient than the time division multiple access (TDMA)-based OMA only when the two users have similar and homogeneous latency constraints. Under heterogeneous latency constraints, the NOMA scheme may be less energy-efficient. To improve the performance of the NOMA schemes, a hybrid transmission scheme which combines both NOMA and TDMA was also presented in [14].

The work in [68] investigated the performance tradeoffs between URLLC and eMBB services under both OMA and NOMA, by considering different interference management strategies such as puncturing, treating interference as noise (TIN), and SIC. It was observed that when analog

fronthauling cloud-radio access network architecture is considered, NOMA achieves higher eMBB rates with respect to OMA while guaranteeing reliable low-rate URLLC communication with minimal access latency. Moreover, NOMA under SIC was able to achieve the best performance while unlike the case with digital capacity-constrained fronthaul links, TIN always outperformed puncturing. However, the best performance was achieved at a price of higher decoder complexity. The performance features of various MIMO schemes were discussed in [55], [70]. In [70], the performance gains of NOMA over OMA were demonstrated.

In [29], [64], [67], multi-tenant networks were considered with eMBB, URLLC and mMTC services. Then, end-to-end network slicing frameworks were proposed for slicing both computing and communication resources across 2-tier multi-access edge computing architecture. In [29], an upper-tier first with latency bounded over-provisioning prevention framework was proposed for optimizing the capacity and traffic allocation. It was shown that the framework in [29] is capable of minimizing the over-provisioning of the network resources while simultaneously satisfying the latency constraint requirements of the URLLC service. Network slicing for Industry 4.0 scenarios with mixed traffic types was considered in [72]. Consequently, a latency-sensitive 5G RAN slicing solution was proposed. The proposed solution focused on allocating the radio resources among slices by considering the rate and latency demands of the applications. It was shown that the solution is capable of effectively improving the capacity of 5G to satisfy the latency requirements of latency-sensitive or time-critical Industry 4.0 applications. Also, in [67], a latency-aware network slicing solution was proposed for a clustered multi-tier multi-tenant 5G network. The solution efficiently allocates radio resources to eMBB, mMTC, and URLLC services by considering the data rate and latency requirements of the services. In [66], it was demonstrated that network slicing can be achieved through the use of AI. Consequently, a FL framework was proposed to predict customer demands of different services in order to effectively allocate appropriate network resources.

F. EDGE CACHING

Data transmission latency can be affected by the access delay between the core network and base station. When edge caching is used, popular content can be stored at the network edge using network edge devices such as cache-enabled base stations and edge servers. Whenever a cached content is requested by a UE, the base station intercepts the request and directly returns the cached content without resorting to a remote server [73]. Consequently, the content delivery latency is reduced, content availability is improved, traffic load on the network path to the remote server is reduced, and the fronthaul efficiency is improved [1], [3], [18], [42], [45], [74]. However, to enable edge caching, more storage and processing power is required at the edge devices. Edge caching schemes for low latency in 5G networks were presented in [1], [3], [30], [73], [75]–[77].

It was demonstrated in [73] that edge caching can offload the network traffic to effectively reduce massive duplication of content downloads and also reduce the communication costs in mobile networks. Furthermore, it was shown that edge caching can satisfy the content requests of mobile users locally to effectively reduce the latency and improve the QoE of mobile users. In [75], three caching transmission schemes were designed to minimize the delivery latency for cache-enabled multi-group multicasting networks. The analysis results in [75] confirmed that if a transmission scheme is designed carefully, caching frequently requested content at the network edge can effectively reduce the delivery latency while reducing the load on the fronthaul links. Furthermore, an integrated content-centric mobile network framework was considered in [76]. Then, an auction based caching strategy was proposed to facilitate edge caching at UEs through D2D communications. It was shown that the proposed framework can guarantee content delivery efficiency by achieving low content access delay and reduced traffic load. In [78], the problem of optimal bandwidth allocation was considered and the average transmission delay was minimized by deploying a greedy algorithm of cooperative edge caching.

The study in [1] highlighted that there are several fundamental tradeoffs for caching in mobile networks including latency versus storage, memory versus rate, memory versus channel state information at the transmitter, storage versus maximum link load, and caching capacity versus delivery rate. To effectively implement edge caching, side knowledge such as the user location, mobility patterns, and social ties can be exploited to decide on which contents to cache and where to cache them [3].

The work in [3], [34], [41], [45], [74] highlighted the advantages of implementing AI for edge caching (edge AI). Edge AI facilitates the interpolation/extrapolation of human activities and predictive caching for reducing the content delivery latency and improving content availability. Edge AI is in alignment with a new machine learning paradigm, called edge learning/edge computing, where learning algorithms are deployed at the network edge for providing intelligent services to the mobile UE [24], [74], [79]. The effectiveness of edge AI is demonstrated in many studied including [24], [34], [79]–[85].

It was pointed out in [34], [41] that mobile edge caching is a challenging decision making problem with unknown future content popularity and complex network characteristics. The problem is caused by the fact that different contents are favored by different users. Also, local content popularity is influenced by the changing membership of the mobile users associated with the edge cache entity. Furthermore, user preferences in contents may vary in different contexts, including personal characteristics, locations, network topologies, and so on. Therefore, future content popularity may be unknown before making the cache decision. To address the challenges, the use of RL mechanisms was considered in [34] because

RL enables agents to deal with decision making problems by learning through interactions with the environment. It was also presented that RL can be coupled with DL to devise DRL solutions. The DRL solutions enable agents to optimize their control in an environment by automatically learning knowledge directly from raw, high-dimensional observations. Then, it was shown that DRL solutions are effective at offloading the traffic and improving content availability and delivery reliability. The Google DeepMind is one of the successful solutions which utilize DRL [34].

Proactive cooperative caching solutions which use DL algorithms to predict user content demand in a mobile edge caching network were proposed in [82]. The first solution was designed to utilize DL to predict the demands for the whole network. The solution proved to be useful when mobile edge nodes had limited computing resources and could not perform the DL algorithms. However, the communication overhead was increased. To address the problem, a second solution was proposed using a distributed DL (DDL) framework. In the DDL solution, the content server only collects the trained models from the mobile edge nodes and updates the global model accordingly. The analysis results in [82] indicated that both DL-based and DDL-based solutions can reduce the service delay and improve the accuracy of prediction, with better performance than other proactive caching algorithms at mobile edge nodes. In [83], an edge learning system was developed for networked intelligent applications to effectively reduce network traffic and inference latency. It was observed that with comparable learning accuracy as a cloud-centric design, an intelligent edge computing solution can significantly reduce the latency and network traffic. As a result, the content acquisition reliability was improved.

An efficient DL cache replacement algorithm for edge networks was proposed in [84]. The algorithm understands the request patterns in individual base stations and accordingly makes intelligent cache decisions. It learns the caching strategy automatically from the request sequence in real-time. As a result, the proposed algorithm is capable of reducing transmission delay and backhaul data traffic. An intelligent video content edge caching framework was proposed in [86] to cater to the massively diversified and distributed caching environment in 5G content delivery networks. The caching framework minimizes both content access latency and traffic cost through the use of a multi-agent DRL (MADRL) solution. The framework ensures each edge is able to adaptively learn its own best policy in conjunction with other edges for intelligent caching. In [30], several DRL edge caching solutions were presented. Then, actor-critic DRL frameworks were proposed for centralized and decentralized edge caching. It was shown that the proposed frameworks are capable of improving the performance by reducing the transmission delay and improving the cache hit rate, adaptation capability, and long-term stability.

In [24], [37], [39], [41], [87]–[90], it was discussed that FL is a promising technique for future intelligent networks due to its superior performance features and added benefits. Edge caching solutions which are based on FL algorithms can guarantee smart models, reduced content delivery latency, improved content acquisition reliability, and improved energy efficiency, all while ensuring preservation of personal data privacy and security. It was shown in [85] that FL can present more interesting performance features than other DDL approaches to support latency-critical applications. Subsequently, the integration of DRL techniques and a FL framework with mobile edge systems was considered, to optimize mobile edge computing, caching, and communication. As a result, it was observed that when a FL framework is used, it can optimize the operations of mobile edge computing systems and ensure personal data privacy.

In [80], a distributed joint transmit power and resource allocation framework for enabling ultra-reliable and low-latency vehicular communication was proposed using a decentralized learning model. The analysis results revealed that FL can enable vehicular users to learn the tail distribution of the network-wide queues locally without sharing the actual queue length samples. As a result, unnecessary overheads were reduced to improve the communication efficiency. The work in [79] focused on the use of a FL framework to design a low-latency multi-access scheme for edge learning. The FL framework presented better latency reduction performance than an OFDM framework. The effectiveness of FL schemes for mobile edge computing was also demonstrated in [88], [91]. In [88], FL edge caching was employed for urban infrastructures and urban informatics, with the consideration of urban vehicular networks. It was demonstrated that FL edge caching can introduce considerable benefits to realize edge intelligence in urban informatics. Furthermore, the FL framework in [88] was able to achieve high accuracy and communication efficiency while preserving the privacy of user data.

G. TSN AND DETNET STANDARDS

The TSN and DetNet standards present effective mechanisms for applications with deterministic and bounded ULL requirements [10]. The key difference between the URLLC in 5G cellular networks and bounded ULL is that, bounded ULL has stricter requirements in terms of latency, packet loss, and delay variation (jitter). Thus, bounded ULL demands absolute time-synchronization and on-time delivery of packets for deterministic and isochronous real-time applications [11]. As a result, the techniques such as packet retransmission become ineffective since the data traffic becomes stale if retransmissions are used. Good examples of applications which require bounded ULL include automotive in-vehicle networking and real-time industrial automation and control systems. For instance, autonomous driving systems include delay-sensitive real-time applications (e.g., machine vision) that cannot tolerate the delay due to retransmissions of lost frames [31]. In such situations, the TSN and DetNet standards can be employed to ensure zero congestion loss, low latency variations (jitter), deterministic ULL, and high link reliability. Thus, the TSN and DetNet standards ensure real-time guarantees, determinism, high fault-tolerance, and clock

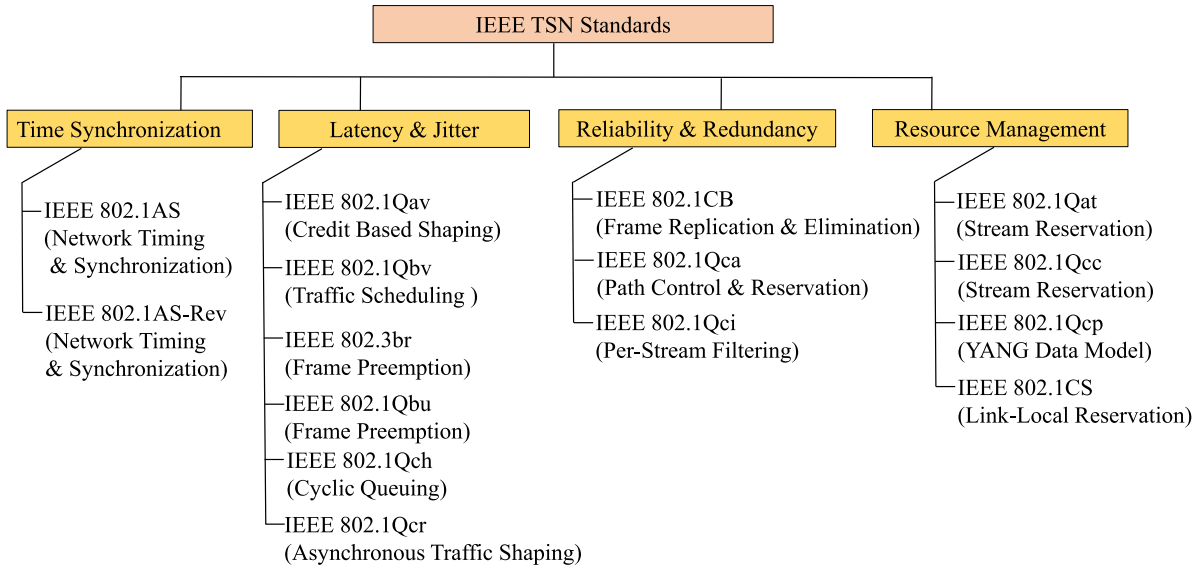

FIGURE 3. Classification of the IEEE TSN standards.

synchronization mechanisms over standard Ethernet [16], [31]–[33], [44], [92]–[96]. Overall, the TSN and DetNet standards enable co-existence of high priority, deterministic, and elastic traffic.

The TSN standard has many (sub)standards as shown in Fig. 3. The standards define how frames belonging to a particular traffic class or having a particular priority are handled by TSN-enabled bridges. Based on the discussions in [16], [95], we classify the TSN standards into four categories based on their objectives. Thus, TSN standards achieve four main objectives:

1) **Time synchronization -** The time synchronization mechanisms aim to provide network wide precise synchronization of the clocks of all entities at layer 2.
2) **Latency and jitter** - The mechanisms for latency and jitter aim to separate traffic into traffic classes and efficiently forwarding and queuing the frames in accordance to the traffic classes. It is worth noting that latency and jitter are the two main QoS metrics for ULL networking.
3) **Reliability and redundancy** - The mechanisms for reliability and redundancy are designed to maintain network wide integrity by ensuring path redundancy and ingress queue policing.
4) **Resource management** - The techniques for resource management are designed to provide dynamic discovery, configuration, and monitoring of the network in addition to resource allocation and registration.

To ensure 5G ULL communications, TSN includes the IEEE 802.1CM standard. The IEEE 802.1CM is designed to provide specific mechanisms such as scheduling, preemption, and synchronization mechanisms to satisfy the 5G fronthaul requirements. Below, we highlight the mechanisms of 802.1CM, 802.1AS, 802.1CB, and DetNet standards. More detailed discussions of the TSN and DetNet mechanisms are presented in [16], [31]–[33], [44], [51], [92]–[97].

The wide range of data rates and latency requirements in 5G and B5G networks require a flexible and scalable fronthaul. In terms of latency requirements, ULL on the order of 100 μs is required for the one-way fronthaul in wireless cellular networks [16]. In the case of throughput requirements, the required throughput is vastly dependent on the application needs, which may vary widely from small amounts of IoT data to large exchanges of media data transfers to and from the cloud (or the fog to reduce latency) [16]. Therefore, the 802.1CM standard specifies TSN profiles for 5G fronthaul [16], [92]. The standard provides bridged Ethernet connectivity for fronthaul networks to support a data rate of 1 Gbps or higher. It defines mechanisms for end stations, bridges, and local area networks to establish Ethernet networks that can support the time sensitive transmissions. Furthermore, it provides specific mechanisms, such as scheduling, preemption, and synchronization mechanisms, to satisfy the latency requirements of the fronthaul traffic [16], [51]. The standard distinguishes its traffic into different classes and a specific fronthaul profile is applied to each class to transport the flows over the Ethernet bridges based on the flow requirements. For network synchronization, the 802.1 CM standard specifies two mechanisms:

1) Packet timing using protocols such as the precision time protocol (PTP) for point-to-point synchronization distribution from a remote common master.
2) Co-located common master for both baseband unit (BBU) and remote radio head (RRH) [16].

To enable real-time communications, the 802.1AS standard employs precise time synchronization which allows all bridges and end systems in the network to synchronize their

local clocks by less than 1 μs precision [33]. In particular, each time-triggered (TT) frame is transmitted according to a static schedule that allows deterministic communication with bounded jitter and end-to-end latency. Furthermore, the static schedule prevents the interference from lower priority frames by preempting the frames in advance of the TT frames arrival. For some critical systems with stringent requirements, the static schedule can provide jitter-free transmission and deterministic end-to-end latency guarantees by isolating the TT frames in every egress port [33].

To ensure reliability, the 802.1CB presents the technique of frame replication and elimination for reliability and fault tolerance. Thus, the 802.1CB introduces seamless redundancy mechanisms for time-sensitive data whereby each data frame is sequenced and duplicated across a redundant link to prevent single point of failure. The mechanism of frame replication and elimination involves two main procedures:

1) Number the sequence of packets and replicate them in the network. Thus, duplicate copies of critical traffic across disjoint network paths.
2) Identify the redundant and eliminate packets at or near the destination. The packets can also be re-replicated or eliminated at various nodes to ensure the failure of a node does not affect the end-to-end packet delivery [31], [92].

Furthermore, the technique of frame replication and elimination removes the need for retransmission since the data traffic becomes stale if retransmissions are used.

The DetNet standard is an extension of the TSN standard with some similar features such as time synchronization, frame replication and elimination for reliability, and the mechanisms for zero congestion loss. The key differences between the TSN and DetNet include the following. While the techniques of TSN are focused on Layer 2, DetNet extends its properties to Layer 3 or even higher layers. Thus, compared to the TSN which are relatively small scale networks, DetNet aims at larger scale networks [94]. Also, to ensure bounded ULL, DetNet defines upper and lower bounds while TSN defines only an upper bound. Moreover, DetNet enables real-time communication between the subnets, provides simple network configuration, and interoperability with the existing IP-based network infrastructure [98].

It was highlighted in [98] that to ensure QoS, DetNet employs the following key mechanisms:

1) **Resource allocation -** The resource allocation technique aims to address the QoS requirements of latency and packet loss. The bounded ULL is achieved by reserving bandwidth and dedicated buffer resources for zero packet loss at each DetNet-aware router. The provision can be released or reused when they are no longer needed.
2) **Service protection** - The service protection technique is used to improve reliability which aims to mitigate or eliminate packet loss by switching the explicit disjoint path after a failure is detected to re-stablish required DetNet service. However, route changes, even after the failure recovery can lead to out of order delivery of the packets. The DetNet service sub-layers includes the packet replication, elimination, and ordering function mechanism which re-orders packets that are received out of order.
3) **Explicit routes** - To improve the deterministic latency, DetNet explicitly uses a specific path to each traffic flow. These flows can be synchronous or asynchronous depending on the application. In synchronous flows, DetNet-enabled endpoint systems use time synchronized clocks. For asynchronous flows, maximum packet size and maximum number of transmissions during the observation interval is used in conjunction with the information from protocol stack to reserve bandwidth for DetNet flows.

More details about the properties and mechanisms of the TSN and DetNet standards were presented in [10], [16], [31]–[33], [44], [51], [92]–[98]. For example, in [32], [93], reinforcement learning-based techniques were proposed for DetNet data forwarding mechanisms. The proposed mechanism in [93] considered the bounded latency requirement, network states, and different resource usages, aiming at optimizing the resource usage of the whole network while satisfying the requirements of as many applications as possible. Therefore, DRL was used to connect the transmission latency with the network states. It was found that the DRL mechanism was capable of adjusting effectively the data transmission for deterministic applications, according to the resource usage of the networks. In [98], edge computing for DetNet was considered. It was presented that providing control from the edge in DetNet has several advantages including improved flexibility, enabled local big-data analysis and processing, and reduced network traffic. The study in [95] evaluated the performance of TSN networks based on the 802.1Qbv and 802.1Qbu mechanisms for carrying real fronthaul traffic. It was observed that 802.1Qbv and 802.1Qbu are capable of protecting high-priority traffic flows even in overload conditions.

On the other hand, the challenges of the TSN and DetNet standards were considered in [10], [16], [94], [96], [97], together with the potential solutions. Here, we highlight some of the challenges. For instance, to guarantee the minimum TSN node transit delay, a node preempts an ongoing low priority frame transmission for transmitting an incoming high priority frame. This can result in a challenge that the latency of the low priority traffic is based on the preemption occurrences which depends on the high priority traffic intensity. If the high priority traffic intensity is significantly higher than the low priority traffic intensity, then the latency of the low priority traffic can be greatly increased [16]. Furthermore, to ensure reliability, the standards replicate the frames of critical traffic across disjoint paths. The frame replication technique may result in inefficient bandwidth utilization. Therefore, the work in [16], [97] presented the techniques to alleviate the limitations. In the case of frame replication and

elimination mechanisms, [16] presented that the packet replication can be controlled based on traffic class and the path information acquired through the TSN stream identification. Also, through a sequence generation function. Moreover, [16] suggested that the integration of protocols and standards can ensure seamless redundancy and fast recovery.

In [97], a machine learning-based intelligent configuration synthesis mechanism was proposed to enable fault detection techniques in order to detect when and where a fault occurs. Subsequently, the intelligent configuration synthesis mechanism replicates the frames only when a link has a higher propensity for failure. In [10], it was pointed out that DetNet can be used to connect remote TSN locations and allow bounded latency multimedia services over the Internet. Also, DetNet can provide mechanisms for the eMBB, URLLC, and mMTC, realizing ''soft'' slicing for assuring the desired service performance without partitioning the allocated resources. However, DetNet supports a relatively low number of services. Therefore, aggregation and queueing are challenging. In [94], it was pointed out that since DetNet aims at large scale networks and to coexist with the Internet, there is a high chance of having misbehaving traffic, which can disrupt the DetNet flows. Therefore, the features and challenges of misbehaving flows were demonstrated and potential solutions were presented.

III. CACHING TECHNIQUES FOR INTERNET OF THINGS

Due to the integration of sensing and computing functions in IoT devices, the IoT devices have become useful in enabling advanced applications such as device-free sensing, ranging from smart buildings to smart cities [21], [25], [74]. Furthermore, the intelligent identification, behavior tracking, and daily management functionalities of the devices allow for IoT systems to be beneficial in various advanced services and mission-critical applications, including smart energy, self-driving cars, ubiquitous e-healthcare, and industrial automation [25], [74]. Therefore, low latency and high reliability are key ingredients for effective IoT device operations in the mission-critical applications. In particular, when multiple operators control the IoT devices remotely.

There exists various techniques and solutions for guaranteeing low latency and efficient content delivery in IoT systems [21], [25], [38], [55], [74], [99]–[103]. Below, we discuss the solutions which are based on the approaches of in-network caching and edge caching. Among many benefits, the solutions are capable of achieving fast content retrieval with low response latency and improved content acquisition reliability.

A. IN-NETWORK CACHING

IP-based networks may not be the best solution for IoT devices because IP-based networks are host-centric and they require the location of content to be identified before the retrieval of the content [26], [35], [104]. To identify the location, every device is assigned an IP-address. As a result, IP-based networks incur some limitations such as poor reliability, poor scalability, and uneven network flows. On the other hand, information-centric network (ICN) has proven to overcome the limitations of IP-based networks. Therefore, ICN is known to become the architecture for the IoT [26].

ICN provides users with access to content by names. Therefore, all content is given a name that does not include references to its location. To access the content, user requests for specific content are routed toward the closest copy of such content, which could be stored in a server, in a cache contained in a network node, or even in another user device. Then, the content is delivered to the requesting user by the network. Subsequently, applications can refer to the content through a name that can potentially contain semantic. Hence, network and applications are allowed to share the same namespace, enabling simpler edge networking, and boosting innovative IoT architectures [105], [106]. The most commonly used naming scheme in ICN is the URL-style naming. For example, a sensor can be named according to its affiliation, such as ''*/BUPT/Building5/Level7/Room7119/Sensor37*'' [107].

One of the most significant features of the ICN is in-network caching. In-network caching considers caching as a basic network functionality and contents are cached in routers of the core network. Therefore, it reduces the content retrieval time for the IoT devices as it brings the required content near the subscriber by caching it on the nearby nodes [26], [35], [36], [108]–[114]. Other benefits of ICN and in-network caching for IoT networks include improved cache hit ratio, reduced hop count, increased content availability and reliability, reduced network traffic, and reduced energy consumption [102], [103], [106], [115]–[117].

The use of the in-network caching technique for IoT was first considered in [108] where the tradeoff between multi-hop traffic load and data freshness was considered. It was observed that in-network caching can provide less load on the network when caching routers are closer to the requesting device than the data sources. Since the work in [108] was presented, numerous designs for ICN caching solutions have been proposed. The strategies in each design consider the features and limitations of the IoT devices. The work in [36], [118] considered that memory is a scarce resource in IoT devices. Therefore, unlike other caching solutions which incur high overheads or require extensive knowledge of the topology, [118] proposed a simple lightweight content caching strategy which makes use of the topology-based heuristics of the existing strategies, but requires no knowledge of the network, and incurs no communications overhead. The proposed strategy uses the caching node centrality or betweenness centrality to decide whether or not to cache content. It was shown that content delivery latency in IoT can be reduced without requiring any setup, global knowledge, or communications overhead when approximate centrality information is used to cache data in the most convenient location, regardless of the topology.

The work in [36] presented the idea that IoT devices may not be able to cache all content comprehensively.

Subsequently, it evaluated a number of in-network caching strategies using real IoT hardware, focusing on the effects of the strategies on content delivery latency. The analysis results revealed that, the topology of the network and the routing algorithms used to generate forwarding information might have a significant impact on the performance of a given caching strategy. In [111], it was pointed out that IoT data is transient and frequently updated by the producer. Therefore, IoT data imposes strict requirements in terms of information freshness. Also, IoT devices are usually resource-constrained with severe limitations on energy resource, memory, and processing power. Therefore, a freshness-aware in-network caching strategy was proposed to monitor the validity of cached contents while improving the hop reduction and server hit reduction ratios. It was indicated that a freshness-aware in-network caching strategy might present very good system performance in terms of response latency, hop reduction ratio, and server hit reduction ratio. Furthermore, it may provide reduced cache costs and significantly improved content validity.

Periodic caching may be the most appropriate strategy for IoT environments such as smart cities [112]. Therefore, a flexible periodic in-network caching strategy was proposed in [112] to reduce the retrieval latency and improve the cache hit ratio. It was pointed out in [26] that in-network caching functionality is very useful for IoT because it reduces the overhead of the publisher with respect to content access. Consequently, an IoT caching strategy was proposed to effectively cache IoT data as well as update and evict content based on its freshness values. The proposed strategy displayed good performance in terms of response latency, hop reduction, and energy utilization. It was revealed in [109] that a collaborative caching strategy can improve the hit ratio of content caching while reducing the average delay and average hops in data-centric networks. Furthermore, a collaborative caching strategy may address the issue of dynamic topology changing, save the cache space, and reduce energy consumption. The benefits of combining ICN and scalable video coding were considered in [102] to improve the performance of wireless video delivery services in ICN. Particularly, the caching problem for scalable videos over ICN in mobile scenarios was investigated. Then, a layered hierarchical caching solution was proposed. It was observed that the proposed solution can apply machine learning techniques to learn user behaviors such as mobility patterns and request distribution. Also, it can perform proactive caching. As a result, it achieves improvements in both transmission latency and cache hit rate.

A pre-caching strategy based on the relevance of smart device request content was proposed in [103]. The strategy was designed to ensure the request and return of the content follow the same path. Subsequent content chunks of requests were pre-cached based on the relevance of content. Furthermore, a sojourn time was set for each content chunk according to its popularity. Observations show that the solution in [103] is capable of reducing content response latency and hop count while improving the cache hit ratio. The tradeoffs between delivery latency and caching cost were considered in [106]. Subsequently, an optimal coded caching solution for joint optimization of delivery latency and energy consumption was proposed for ICN-based 5G D2D networks. A D2D-based caching scheme was adopted in [115] to reduce the downloading latency while guaranteeing reliable data delivery in social IoT applications. A content sharing-oriented matching algorithm was considered with projecting social characters onto physical links. A near-optimal solution was found by using a heuristic searching algorithm. As a result, a low-latency and high-reliability achieving caching solution was devised, specifically for social IoT application scenarios.

The limitations of on-path caching strategy for ICN were considered in [116]. It was pointed out that although on-path caching can increase the content cache hit ratio in some cases, it can also result in redundant use of the cache memory and reduced overall efficiency of the cache strategy. Therefore, a cluster-based efficient caching mechanism was proposed to address the problems of on-path caching mechanism. Furthermore, a popularity-driven content replacement mechanism was considered as an architecture for efficient caching and fast content access mechanisms for data-intensive IoT applications. It was observed that when content popularity is considered during cache decisions, the content transfer time and packet loss ratio can be reduced while increasing the content availability. It was also observed in [117] that in mobile hybrid ICN-IoT, both mobile devices and static femto access points might be equipped with finite-size cache space. Then, an order-optimal cache allocation strategy can be applied to optimize the performance. Highly popular content objects can be served by the mobile devices while the rest of content objects are served by the static femto access points. As a result, the best throughput–delay tradeoff can be achieved to optimize the performance.

In [119], an edge-device-based ICN caching scheme was proposed with the aim to achieve comparable performance to in-network caching schemes. The scheme employed a lightweight collaborative caching framework. It allows edge devices to cache passing contents while routers simply maintain cache indexes and use them to redirect user requests towards nearby cache locations. The design of the scheme ensures efficient content distribution and achieves good performance in terms of response latency, server load, and bandwidth consumption of the links.

B. EDGE CACHING

Edge caching and edge computing for IoT systems were considered in [25], [38], [120]–[125]. When edge caching and edge computing are implemented, IoT devices can offload their intensive tasks to edge nodes to allow the tradeoffs between communication and computing. As a result, performance enhancement can be achieved. The work in [120] considered the issue of how to efficiently use the limited caching resources on the IoT devices to place contents so as to improve the hit probability. It highlighted that in some scenarios, caching the most popular content is not the

optimal strategy. Subsequently, an optimal probabilistic caching strategy was utilized to place contents with different sizes in heterogeneous IoT networks while considering the serving capacity constraint on the caching devices. In [25], various scenarios of IoT-based smart cities were considered. The main objective was to find out how to allocate edge resources for average service response time minimization while running multiple smart city services over a large number of IoT devices and satisfying the capacity constraints of edge servers. Subsequently, the benefits of edge computing were assumed. Computing and storage resources were employed at the proximity of end IoT devices and applications were deployed in distributed edge servers.

The work in [121] considered application scenarios where videos captured by surveillance cameras are required to be delivered to remote IoT servers for video analysis. The transmissions to the remote IoT servers usually involve long-distance transmissions of the large volume of video chunks which may cause congestions and delays due to limited network bandwidth. Therefore, an edge computing framework was proposed to enable cooperative processing on resource-abundant mobile devices for delay-sensitive multimedia IoT tasks. The framework considered group formation and video–group matching algorithms to maximize the human detection accuracy within the video task deadline. It was observed that the proposed solution in [121] was capable of reducing the delays of processing the task at the camera and uploading the data to the server. Furthermore, other advantages such as reduced costs of uploading the large volumes of data and easy deployment were observed.

Edge caching for industrial IoT application scenarios was assumed in [122]. Consequently, the challenges of contents dissemination in characteristic factory automation scenarios were addressed by using moving industrial machines as D2D caching helpers. The main goal was to improve the reliability of high-rate mmW data connections. Therefore, a novel mobility-aware methodology was constructed to help develop predictive mode selection strategies based on the anticipated radio link conditions. The findings in [122] revealed that predictive solutions that employ D2D-enabled collaborative caching at the wireless edge can lower content delivery latency and improve data acquisition reliability. In [126], a study was done to maximize the success probability of content sharing in social IoT through the use of multi-hop cooperative caching and communication. It was observed that multi-hop cooperative caching can improve reliability in terms of content sharing in D2D-enabled social IoT. It was shown in [127] that for users within the same coalition, significant delay cost reduction can be achieved through the use of selective file caching in edge devices and cooperative file sharing. D2D-enabled caching via coalitional game was devised to effectively reduce the average delay cost of users. Other interesting solutions for edge caching may be available from content delivery network providers such as [128], [129].

Learning algorithms such as ML/DL have shown to be effective in enabling caching-supported IoT networks [74], [113], [124], [130], [131]. Therefore, in [74], the issue of massive content access in mobile networks was discussed to support rapidly growing IoT services and applications. The use of edge caching was considered and a FL-based DRL cooperative edge caching framework was proposed. It was shown that caching the requested contents in edge nodes can improve the efficiency of content access compared to excessive downloading via backhaul links. The analysis results in [74] revealed that a FL-based DRL cooperative edge caching framework can achieve improved performance in terms of the average delay, hit rate, performance loss, and the backhaul traffic offload. It was suggested that the proposed framework might be useful in new systems such as the recently launched Apple Edge Cache service. The service enables the delivery of Apple content services directly to the equipment within content provider partner networks [74].

In [130], DL for IoT was used in edge computing environment to improve the learning performance and reduce the network traffic. Furthermore, the benefits of using DL over ML for IoT applications were discussed in [130]. The benefits include a better performance with large data scale because many IoT applications generate large amounts of data for processing. Also, DL takes much less time to inference information than traditional ML methods. The work in [124] considered the problem of network congestion caused by the traditional cloud computing model. Subsequently, [124] proposed an AI-enabled edge computing framework for heterogeneous IoT architecture. The framework implemented a joint optimization mechanism of edge computation, caching, and communication. As a result, it achieved a lower average delay than in the traditional cloud computing model while increasing the number of concurrent users.

In [113], it was discussed that it is important to incorporate the future popularity of content into the cache decision making although many fundamental questions about IoT data popularity and related popularity-based caching may not have obvious answers. Therefore, the questions and answers were explored and a popularity-based caching solution for the IoT devices was proposed. The proposed solution employed a data-driven caching model which utilizes all the historical request records and an offline deep neural network to generate the accurate popularity model. The popularity model was used in the online caching algorithms. It made the cache decision based on the temporal locality of the request, long term regularity, the features of data itself, and the popularity of other similar data. The experiment results in [113] revealed that a data-driven caching model can achieve reduced content delivery latency and IoT edge traffic while significantly improving the cache hit ratio.

The benefits of caching transient IoT data at the edge nodes were demonstrated in [131] through the use of a DRL caching solution. The DRL solution was able to intelligently perceive the environment. It automatically considered a caching policy according to history and current raw observations of the environment, without any explicit assumptions about the operating environment. Consequently, a balance between

the communication cost and the loss of data freshness was achieved. The freshness of the information was also considered in [132]. Then, a RL-based scheduling algorithm was devised to optimize the age of information and ensure URLLC. In [125], DRL was coupled with FL to effectively employ edge computing in IoT environment. It was observed that a FL-based DRL solution can achieve reduced transmission costs between IoT devices and edge nodes in dynamic IoT systems.

IV. LIMITATIONS OF THE ENABLING TECHNIQUES FOR LOW LATENCY AND RELIABLE COMMUNICATIONS

Table 2 presents a summary of limitations of the enabling techniques. Furthermore, Table 2 includes the potential approaches to mitigate the limitations. It shows that although the techniques such as dynamic multiplexing and enhanced power control mechanism can be used to mitigate some of the limitations, the use of the techniques can also result in other limitations. For example, although dynamic multiplexing can address the limitations of latency sensitive scheduling schemes, it causes increased computational load on the eMBB UE. Furthermore, relying upon the transmission of data replicas to achieve a target reliability level in grant-free access can result in inefficient network resource utilization. Moreover, it was observed in [53] that the transmission of replicas for each packet may cause additional delays, may be inadequate, and it cannot support URLLC for services with sporadic traffic, mainly due to the impact of self-collisions. Also, although the enhanced power control mechanism can mitigate the limitations of dynamic multiplexing, the mechanism may not be suitable when the URLLC UE is already power-limited. This is mainly because the mechanism often requires the URLLC UE to boost its power by a significant amount. Two methods which can alleviate the limitations of the enhanced power control mechanism are enhanced open-loop and enhanced closed-loop power control mechanisms [22].

It was highlighted in [66], [133] that an important limitation for network slicing is how to effectively ensure traffic isolation and personal data privacy between network slices/services. Therefore, it was proposed in [133] that when slicing-enabled multi-access edge computing (MEC) framework is used, the radio network information service and location service APIs should be modified to include a slice identifier of UE in addition to the UE identifier. By doing so, it becomes possible to restrict the applications to access only information on UEs of their slice. A FL framework was presented in [66] to predict the customer demand of each service class in order to allocate appropriate network slices. The framework allows the exploitation of personal user data while preserving the privacy of sensitive information. The limitation of reduced personal data privacy in edge caching technique was also considered in [27] and a privacy-preserving edge computing scheme was presented. It was shown in [116] that in in-network caching technique, although on-path caching can improve the delivery reliability, it can also result in redundant use of the cache memory to reduce the overall efficiency of the cache strategy. Then, it was demonstrated that a cluster-based efficient caching mechanism can be employed to address the drawbacks of on-path caching mechanism.

On the other hand, in traffic flows with deterministic behavior and bounded ULL, the TSN and DetNet standards present effective mechanisms. However, the TSN and DetNet incur several challenges as highlighted in [10], [16], [94], [96], [97]. For example, the latency of low priority traffic is based on the preemption occurrences which depends on the high priority traffic intensity. If the high priority traffic intensity is significantly higher than the low priority traffic intensity, then the latency of the low priority traffic can be greatly increased. It was presented in [16] that TSN requires flexible and dynamic priority allocations to ensure bounded end-to-end latency for lower priority traffic. Furthermore, when frame replication mechanism is used to improve the delivery reliability, the technique can result in inefficient network resource utilization. In [16], it was suggested that some integration of the protocols and standards can ensure seamless redundancy and minimize network congestion in TSN. To ensure a balance between bandwidth utilization and degree of packet replication in DetNet, [16] proposed a reverse packet elimination mechanism in which the destination node triggers an instruction to the nodes in the reverse path to apply a packet drop action. The proposed mechanism is different from the existing implementation which discards the redundant packets when they eventually arrive at the destination. Furthermore, to enhance the bandwidth utilization, a machine learning-based intelligent configuration synthesis mechanism was proposed in [97] to enable fault detection techniques in order to detect when and where a fault occurs. Subsequently, the intelligent configuration synthesis mechanism replicates the frames only when a link has a higher propensity for failure.

V. PERFORMANCE FEATURES OF AI-ENABLED EDGE CACHING SCHEMES

As shown in the discussions above, edge caching techniques are widely explored as effective solutions for achieving reduced content delivery latency and improved content acquisition reliability. Furthermore, edge caching, edge computing, and edge AI have great potential in 5G and B5G networks to enable the emerging technologies that will revolutionize our day-to-day operations [9], [28], [135]. For instance, edge caching and edge AI will have potential to revolutionize the automobile industry by integrating fully autonomous vehicles, IoT, and the environment. As a result, major investments have been made by world leading businesses focusing on the benefits of edge computing and edge AI [28]. Furthermore, it is assumed that most 5G and beyond networks will be based on decentralized and infrastructureless communication to enable devices to cooperate directly over D2D spontaneous connections [136]. The networks are designed to operate when needed without the support of a central coordinator or

TABLE 2. Summary of limitations of the enabling techniques.

Technique	Limitations	Mitigation of the limitations
Configurable subcarrier spacing	• Increased control overhead due to use of sTTI. • Inefficient resource utilization.	• Grant-free access is used in the uplink transmissions.
Grant-free access	• Increased number of packet collision events. • Reduced communication efficiency.	• Data replication. • Grant-free SCMA such as SCMA with sparse spreading codebook design. • Grant-free NOMA with advanced receivers such as expectation propagation algorithm or message passing algorithm.
Latency sensitive scheduling schemes	• Inefficient resource utilization. • Increased control overhead in the case of dynamic reservation scheme.	• Dynamic resource sharing through dynamic multiplexing of hybrid services such as URLLC and eMBB.
Dynamic multiplexing	• When URLLC and eMBB services coexist, and when uplink preemption mechanism is used, it imposes higher requirements on the eMBB user equipment processing capability.	• Enhanced power control mechanism at the URLLC UE where base station can schedule an urgent URLLC transmission on top of an ongoing eMBB transmission, and indicate the URLLC UE to boost its power to guarantee reliability. Thus, eMBB and URLLC transmissions are super-positioned on the overlapping resources.
Network slicing	• In heterogeneous latency and reliability constraints at downlink users, SIC technique may not always be feasible since there exists situations where none of the receivers can perform SIC and decode messages of other users. • Under heterogeneous latency constraints, NOMA schemes may be energy-inefficient.	• Hybrid transmission schemes which combine NOMA and TDMA mechanisms.
	• Challenging to ensure traffic isolation and personal data privacy between services.	• When slicing-enabled MEC is used, modify radio network information service and location service APIs to include a slice identifier of UE in addition to the UE identifier. • AI with FL algorithms.
Edge caching	• Increased amount of storage and processing power is required at the cache-enabled edge devices.	• AI-enabled cache decision algorithms.
	• When centralized edge caching system is used, the privacy of personal data is reduced.	• FL caching algorithms.
	• When content popularity prediction is based on historical patterns and social patterns, the technique becomes less adaptive and does not fit well in highly dynamic and heterogeneous application scenarios.	• Multi-agent DRL and FL to adapt the massively distributed dynamics and diversities and achieve collaborative intelligence.
In-network caching	• When on-path caching strategy is used, it can result in redundant use of cache memory and reduced overall efficiency of the cache strategy.	• Cluster-based efficient caching mechanism.
TSN and DetNet	• In TSN, latency of the low priority traffic is based on the preemption occurrences which depends on the high priority traffic intensity. If the high priority traffic intensity is significantly higher than the low priority traffic intensity, then the latency of the low priority traffic can be greatly increased.	• Flexible and dynamic priority allocations can ensure bounded end-to-end latency for lower priority traffic.
	• The mechanism of frame replication across disjoint paths may cause inefficient network resources utilization.	• Reverse packet elimination mechanism. • Integration of protocols and standards to ensure seamless redundancy. • Machine learning-based intelligent configuration synthesis mechanism.

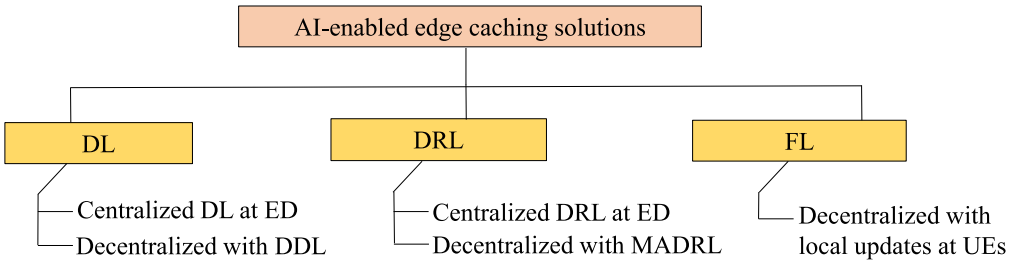

FIGURE 4. Classification of AI-enabled edge caching solutions.

with limited support for synchronization and signaling. Thus, the networks assume content servers and UEs that manage part of the computing tasks locally to ensure fast feedback and low latency [136]. Considering these trends, research activities are now focusing on fully decentralized AI-enabled edge caching techniques. However, edge caching is a challenging decision making problem with unknown future content popularity and complex network characteristics [34]. To ensure optimal cache content placement, caching agents are equipped with AI-enabled learning models to predict the content popularity distribution and indicate what contents are likely to be requested by users.

Therefore, in this section, we discuss the strategies and performance features of the state-of-the-art AI-enabled edge caching solutions which were presented in sections II and III above. We classify the solutions into three categories according to the learning algorithm of the caching agents. Thus, we classify the solutions into DL-based, DRL-based, and FL-based edge caching solutions, as shown in Fig. 4 and Table 3. Hence, we present the features and performance characteristics of the DL-based, DRL-based, and FL-based edge caching frameworks. Specifically, we discuss the learning models of the caching agents which enable the caching agents to understand the user content request patterns and accordingly make intelligent cache decisions.

A. STRATEGIES AND LEARNING MODELS

In general, edge caching-enabled systems can be classified into centralized or decentralized/distributed caching systems [30]. Centralized edge caching systems are equipped with a cache-enabled edge device (ED) such as base station which acts as a content server for all UEs in its coverage. When the ED receives content requests from a UE, it first checks if the contents requested by the UE are cached locally. If the requested contents are available in the ED cache, then the ED can transmit the contents to the corresponding UE without requesting the contents from a remote cloud server [124]. Therefore, it reduces the latency and core network traffic. To avoid requesting content as much as possible, the ED updates its cache according to the user preferences. The user preferences may be affected by factors such as the user location, time of the day, current activities, and the type of the UE. Therefore, each time a request arrives, the ED first checks if the requested content is available locally so as to decide how to serve the UE. Then, the ED decides whether or not to update its cache based on the cache algorithm. The ED uses AI-enabled learning/training models to allow cognitive computing and help the devices understand the request patterns and accordingly make intelligent cache decisions [30], [84]. However, due to bandwidth, storage, and privacy concerns, centralized edge caching systems are often impractical [91]. The centralized caching algorithms may result in overconsumed network resources during the training and data transmission processes [74]. The effects of centralized caching schemes can become severe in dynamic and heterogeneous large-scale systems. The centralized edge caching frameworks can be realized through DL or DRL algorithms as shown in Fig. 4.

In decentralized caching frameworks, the systems may contain multiple cache-enabled EDs and UEs. Each ED is capable of serving the UEs in its coverage. The EDs serve the UEs from the cache when the requested contents are available locally. For the contents not cached locally, a request is generated by the ED to retrieve the content from the remote cloud server. Based on the request patterns, the EDs learn about the user preferences by using AI-enabled learning algorithms and accordingly make intelligent cache decisions [30]. Furthermore, the AI-enabled learning algorithms allow the systems to intelligently utilize the collaboration among the devices to exchange the learning parameters for better training and inference. Therefore, the frameworks are enabled to carry out dynamic system-level optimization and application-level enhancement while reducing excessive system communication load [85]. Decentralized caching schemes have shown to be able to address many challenges of centralized edge caching frameworks, as shown in the discussions below. The decentralized edge caching frameworks can be realized through DL, DRL, or FL algorithms as shown in Fig. 4. We discuss the DL-based, DRL-based, and FL-based caching schemes below.

1) DL-BASED EDGE CACHING

DL is a popular approach for edge caching in wireless networks. The increased popularity of DL in edge caching and wireless networks is mainly due to the following reasons:

- There is a huge amount of data traffic generated on the Internet. For instance, rapid advancement of IoT and

TABLE 3. Performance features of DL, DRL, and FL edge caching frameworks.

Technique	Limitation	Example solution	Objective	Learning model	Performance
DL-based edge caching	• Traditional DL-based optimization and prediction schemes take a long running time of recursions for converging to the optimal model. • In centralized schemes, sending streams of raw training data to server can increase network traffic and energy consumption. • Most schemes cannot handle non-IID data or privacy preservation issues.	DLs [84]	• Reduce latency and backhaul network traffic for 5G mobile video streaming.	• Decentralized DL algorithm.	• Lower latency than LRU and LFU. • Higher cache hit rate than LRU and LFU.
		DLs2 [82]	• Reduce service delay for UEs and error of content demand prediction.	• Centralized DL algorithm.	• Lower latency than LRU and LFU. • Higher cache hit rate than LRU and LFU. • Higher latency than DDLs. • Lower cache hit rate than DDLs.
		DDLs [82]	• Reduce service delay for UEs and error of content request prediction while preserving privacy of UEs data.	• Decentralized DDL algorithm.	• Lower latency than LRU, LFU and DLs2. • Higher cache hit rate than LRU, LFU and DLs2.
DRL-based edge caching	• Requires intensive computation capacity for finding optimal model particularly in large-scale data. • Achieves reduced performance when UEs and network states are heterogeneous. • In large-scale data with massive UEs, centralized DRL schemes incur increased traffic on uplink wireless channels. • In large-scale data with massive UEs, it is challenging to perform decentralized DRL due to relatively weak computation capability of UEs. Also, it takes long time to train the DRL agent. • Decentralized DRL increases the energy cost at the UEs. • Most schemes cannot handle unbalanced and non-IID data or privacy preservation issues.	DRLs [131]	• Minimize communication cost and loss of data freshness in IoT applications.	• Centralized DRL algorithm.	• Significantly lower latency than LRU and LFU. • Significantly higher cache hit rate than LRU and LFU. • Higher latency than MADRLs. • Lower cache hit rate than MADRLs.
		MADRLs [86]	• Minimize content access latency and traffic cost in diversified 5G video streaming environment.	• Decentralized MADRL algorithm.	• Significantly lower latency than LRU and LFU. • Significantly higher cache hit rate than LRU and LFU. • Lower latency than DRLs. • Higher cache hit rate than DRLs.
		DRLs2 [30]	• Improve cache hit rate in media-enabled applications.	• Centralized DRL algorithm.	• Higher cache hit rate than LRU and LFU but lower than MADRLs2.
		MADRLs2 [30]	• Reduce latency and improve cache hit rate in media-enabled applications.	• Decentralized MADRL algorithm.	• Lower latency than LRU, LFU, and DRLs2. • Higher cache hit rate than LRU, LFU, and DRLs2.
FL-based edge caching	• When traditional algorithms such as FedAvg is used, FL suffers from a large number of communication rounds to convergence with non-IID datasets. Also, has high communication overhead.	FedDRLs [74]	• Reduce latency, performance loss, and backhaul traffic while improving hit rate in IoT systems.	• FL-based DRL algorithm.	• Significantly lower latency than LRU and LFU. • Significantly higher cache hit rate than LRU and LFU. • Latency and cache hit rate are comparable to a centralized DRL scheme.
		FedDRLs2 [85]	• Make mobile communication system cognitive and adaptive, reduce network traffic, and achieve near-optimal performance with low overhead of learning.	• FL-based DRL algorithm.	• Significantly higher cache hit rate than LRU and LFU. • Cache hit rate is comparable to a centralized DRL scheme.
		FedDRLs3 [125]	• Reduce transmission costs between IoT devices and EDs.	• FL-based DRL algorithm.	• Transmission time is comparable to a centralized DRL.

social networking applications results in an exponential growth of the data traffic generated at the network edge [34], [79], [91], [137].
- The technique of DL has the ability to extract accurate information from raw data obtained from devices such as IoT devices which are deployed in complex environments [130].
- DL has the ability to learn (train) models using large amount of data [91].
- Due to its multilayer structure, DL is appropriate for the edge computing environment since it is capable of offloading parts of learning layers in the edge and then transfer the reduced intermediate data to the centralized cloud server [130].
- DL can extract new features automatically for different problems [130].

Thus, DL enables the integration of more intelligent functions in order to optimize the network operations and ensure in real-time, different needs of the emerging wireless applications [39]. With DL, devices are able to intelligently control their environment as well as proactively taking more adequate actions by learning and predicting the dynamic evolution of various network features such as the traffic pattern, content requests, mobility distribution, communication channel dynamics, and user context. For example, DL can precisely predict the home electricity power consumption using the data collected by smart meters, with the objective of improving the electricity supply of the smart grid [130]. It was shown in [82] that DL algorithms can perform better than other proactive caching algorithms. However, traditional DL schemes are cloud-centric and they require a stream of raw training data to be sent and processed in a centralized server [39], [74].

The process of sending streams of raw training data to a centralized server can result in several challenges, including slow response to real-time events in latency sensitive applications, excessive network communication resource consumption, increased network traffic, high energy consumption, and reduced privacy of training data [39], [43], [79], [87], [124], [137], [138]. Therefore, traditional DL frameworks may not be suitable in application scenarios with large-scale data that require low latency, efficiency, and scalability [87], [137]. Furthermore, it may be impractical to employ traditional DL frameworks in applications with privacy sensitive training data [39], [87]. Hence, a lot of research work is being done to implement decentralized learning frameworks, in which all the private data is kept where it is generated and only locally trained models are transferred to a central entity [24], [37], [39], [87]–[90].

As an example, the centralized DL framework proposed in [82] was able to predict user content request with high accuracy and deal with dynamic user requests based on the most up-to-date information. The framework proved to be useful in reducing the service delay and improving cache hit rate. However, the communication overhead was high. Also, there were concerns about the privacy of personal data. To address the problems, a DDL framework was proposed. The key difference between the DL and DDL frameworks was that, in the DL framework, all dataset was sent to a centralized server for the learning process to allow the devices to cache the globally most popular contents based on the global knowledge in the network. On the other hand, for the DDL framework, the dataset was learned locally with some exchanged information shared among the devices. Thus, the learning process was distributed and the caching was based on local UE's most popular content. The global model at the server was updated according to the local trained models at the UEs. It was shown that the DDL framework achieved better results than the DL framework in terms of latency and cache hit rate. It was also shown that the DDL framework was able to learn the dataset faster than the DL framework as the number of EDs increased. However, DDL frameworks are not effective at handling not independent and identically distributed (non-IID) data [125], [139]. Decentralized learning schemes can be used to address the challenges of DL frameworks. In particular, the technique of FL can be employed to alleviate the challenges of DL frameworks [43], [125], [138], [140].

2) DRL-BASED EDGE CACHING

DRL edge caching schemes are not based solely on the content popularity information. The schemes employ DRL agents to observe enough features of the environment and ensure accuracy of the cache algorithms [30]. Thus, DRL caching systems are designed to be context-aware by intelligently exploiting context information of the users and statistical traffic patterns [34]. Context-aware caching algorithms are useful because the operating environment in mobile edge caching can be complex and dynamic. For example, user preferences in terms of content can be affected by the user contexts, such as location, personal characteristics, and device diversity which makes complicated patterns. Furthermore, the responses to user requests from EDs can be affected by the network conditions such as network topology, wireless channel, and cooperation among the EDs. Therefore, context awareness allows the caching agents to be aware of their operating environment so that they can make the right decisions when selecting appropriate content to be cached in the limited storage space at the right time, for maximizing the caching performance [34]. Moreover, due to the dynamic characteristics of wireless networks, the use of RL techniques ensures that the caching agents learn maximal rewards while interacting with the environment on trial and error basis [41], [141]. Therefore, RL schemes are employed to enable learning agents to deal with decision making problems by learning through interactions with the environment. Furthermore, the integration of DL and RL capacitates the caching agents to optimize their control in an environment by automatically learning knowledge directly from raw, high-dimensional observations [34].

As an example, Fig. 5 illustrates a DRL process for edge caching in IoT applications. In the process, the caching agent is directly trained with raw high-dimensional observation

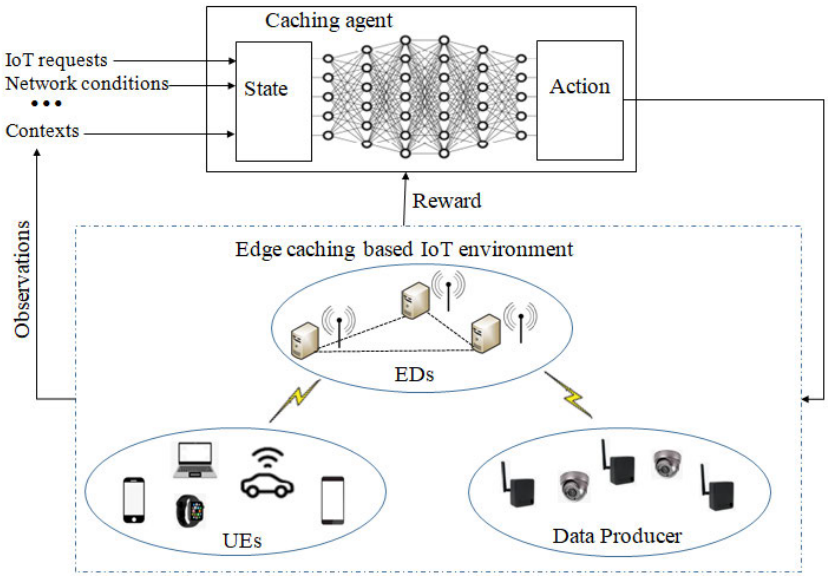

FIGURE 5. An illustration of DRL process for caching IoT data.

signals obtained by observing the state of the environment [131]. The signals can include user requests, context information, and network conditions. The agent is fed with the signals and outputs the action for the caching policy. Thus, according to the output, the agent selects a caching action and observes the reward of performing that action. Accordingly, the agent can adjust and improve its model based on the reward. The process is repeated until the agent model approaches the optimal model [131]. The process is similar in other application scenarios such as the Internet of connected vehicles (IoCVs) scenarios considered in [135]. However, in the IoCVs scenarios, the observation signals include other information such as the capabilities of the road side units and servers, parameters for computing tasks and requested content, and the vehicular mobility information such as the location and velocity.

Traditional DRL frameworks usually employ centralized learning schemes with a single caching agent and they are more suited for small scale network systems [30], [85], [86]. This is mainly because using one central agent for caching decision is not scalable and restricts the practical usage of the edge caching systems. Furthermore, a centralized DRL algorithm has limited functionalities. For example, with a centralized DRL caching algorithm, it may not be possible to employ multiple EDs which would employ unique caching strategies based on their corresponding user request and content access patterns. This can be a limitation considering that the content access patterns is significantly different among different EDs, especially for massively distributed, dynamic, and complicated environments. Moreover, since the DRL solutions employ algorithms that train agents by sharing their raw data, it can result in large amount of network resource consumption during the data transmission process [74].

The performance of the DRL frameworks can be affected by various factors such as the location where the DRL agent is trained, the form in which the training data is

gathered, and how the update process of the DRL agents is proceeded and collaborated [85], [125]. For example, if the DRL agent is trained on the ED or remote cloud server, the following shortcomings may occur:

- In the case of IoT application scenarios with massive UEs, the training data may be in large quantities. Transmitting massive training data from the IoT devices to the ED will increase the burden on the wireless channels.
- May jeopardize the privacy of personal data since the uploaded training data might be privacy-sensitive.
- If the training data is transformed for privacy protection, the proxy data received by the EDs become less relevant [85], [125].

On the other hand, if the training of the DRL agent is done in a distributed approach by training the DRL agent on the UEs, the following shortcomings may occur:

- Consumes long time or even impossible to train each DRL agent well from scratch.
- Extra energy wasting is caused by the standalone training of separate DRL agents [74], [85], [125].

Also, many DRL frameworks are not able to handle unbalanced and non-IID data or cope with the privacy issues [85]. Moreover, the schemes achieve reduced performance when UEs and network states are heterogeneous [9], [85]. To address the shortcomings, it is practical to adopt distributed DRL training through the use of FL. The superiority of the FL-based approaches over DRL approaches was demonstrated in [74], [85], [125].

When a dynamic and heterogeneous large-scale IoT application scenario similar to the scenario in [125] is considered,

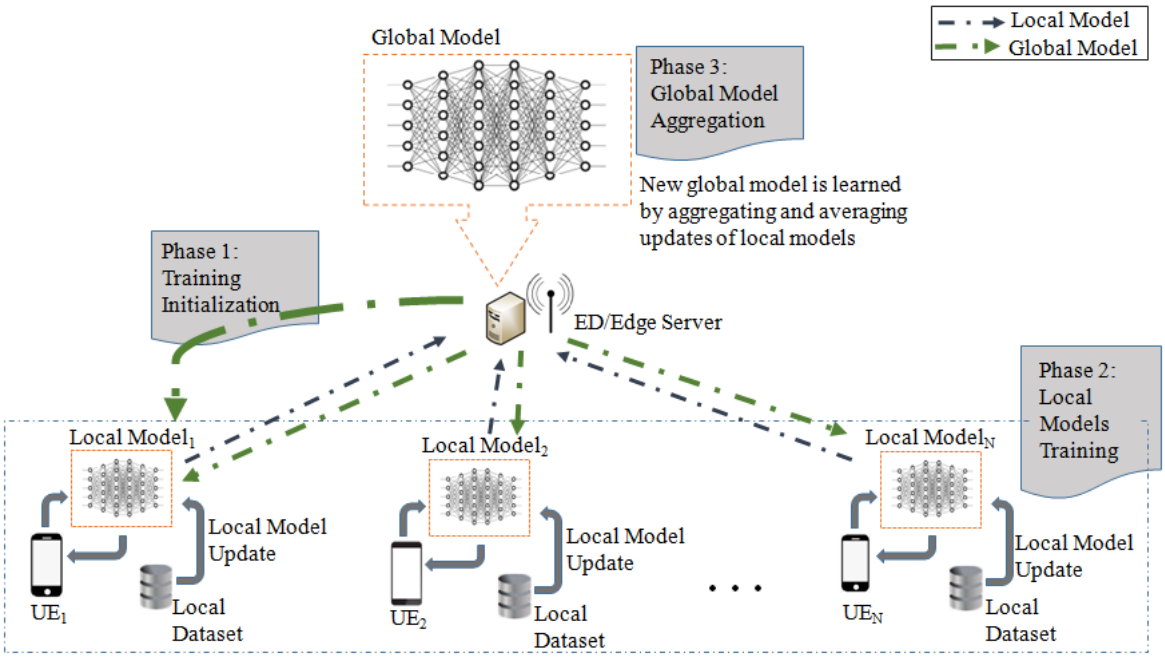

FIGURE 6. An illustration of FL process.

the DRL techniques are efficient in finding optimal training policy in the dynamic edge system but require abundant computation resources [74], [125]. Therefore, in situations with diverse, massively distributed dynamics, and complicated conditions such as spectrum sharing in vehicular network and traffic signal control, DRL can be integrated with multi-agent learning to expand the learning capacity through MADRL [86]. The MADRL approach is decentralized and allows each ED to adaptively learn its own best policy in conjunction with other edges and is capable of achieving collaborative intelligence for edge caching environment that needs cooperation and competition [86]. Furthermore, MADRL schemes are capable of solving the problems of centralized DRL. In [30], [86], it was shown that MADRL frameworks are capable of achieving better performance than centralized DRL frameworks in terms of content access latency, cache hit rate, and traffic cost.

3) FL-BASED EDGE CACHING

FL is an emerging technique that trains distributed ML models with the locally distributed datasets, without sending raw data to a centralized server [88], [142]. The technique was first presented by Google in 2016 [143]. FL is achieved by exchanging only model parameters learned locally at each UE [85], [141], [143], [144]. Furthermore, FL presents unique and attractive features that facilitate collaborative training of local learning models without compromising the privacy of training data [9], [24], [41], [140]. In general, FL presents an efficient training pattern which involves three main phases as demonstrated in Fig. 6. In phase 1, the global model on the server initializes model parameters and then all the UEs download the shared global model. In phase 2, each UE trains the model on its local data independently. Thus, each UE computes an individual update based on its local dataset. In phase 3, all local trained models are uploaded to the server via a secure protocol tunnel and are aggregated to learn a new global model. Then, the new global model is sent back to the UEs [37], [74], [125], [138], [143]. The learning process is iterated for many communication rounds until the global model is able to converge to the global optimal, a threshold level is achieved, or a desirable training accuracy is achieved [24], [141], [143]. To ensure reduced communication cost, FL frameworks employ selective communication such that only the important or relevant updates are transmitted in each communication round [37]. Only the model parameters or gradients are transmitted instead of the raw data. Hence, the FL frameworks employ algorithms that ensure the information that needs to be uploaded is the minimal update which is necessary to improve the accuracy of the global model [37]. When real-time decisions are made locally at the UEs, latency is significantly reduced compared to when decisions are made at the server before transmitting them to the UEs. Also, due to the distributed learning models, FL algorithms have shown to be more practical in large-scale edge caching systems [9].

According to [139], FL edge caching frameworks are suitable for problems with the following properties:

- Training on real-world data from mobile devices provides a distinct advantage over training on proxy data that is generally available in the server.
- The data is privacy sensitive or large in size (compared to the size of the model), so it is preferable not to

log it to the server purely for the purpose of model training.
- For supervised tasks, labels on the data can be inferred naturally from user interaction.

We highlight the two key unique features of FL as compared to centralized learning as follows:
- The learned model is shared between the UEs and the cloud server. However, the training data which is distributed on each UE is not available to the cloud server.
- Instead of the cloud server, the training of the learning model occurs on each UE. The cloud server receives the local gradients and aggregates these gradients to obtain a global gradient and then send the global gradient back to all the UEs [138].

Furthermore, we highlight the unique properties of FL as compared to other decentralized ML algorithms as follows:
- **Non-IID data** - The training data on a given UE is typically based on the usage of the device by a particular user, the wireless environment the UE experiences, the computation capability of the UE, and the energy consumption of the UE. Hence, any UE local dataset will not be representative of the training data of all UEs [85], [139]. In FL, the challenge of non-IID data can be met by merging the updates of the models by using the FederatedAveraging (FedAvg) mechanism [85].
- **Unbalanced data** - Some users will make intensive use of a service or app than others. This can result in varying amounts of local training data. Furthermore, some UEs may have more computation tasks to be handled and some may experience more states of mobile networks. This can result in unbalanced training data among the UEs [85], [139]. Also, this challenge can be handled by the FedAvg mechanism [85].
- **Limited communication** - UEs are frequently and unpredictably offline, on slow or expensive connections, or they are allocated with poor communication resources [85], [139]. However, in FL, using additional computation could decrease the consumption of communication rounds needed to train a model. Moreover, FL only asks a part of UEs, in one round, to upload their updates. As a result, FL handles the situations where UEs are often unpredictably offline [85].
- **Privacy preservation** - FL has the ability to collaboratively train a learning model on their individually gathered data, without revealing their privacy-sensitive data to a centralized server [88], [138], [144], [145]. Also, the information that needs to be uploaded to the server is the minimal update necessary. This feature is particularly useful since in many real-world applications, datasets from UEs are often privacy-sensitive and it is often difficult for servers to guarantee building models of high quality. The techniques of secure aggregation and differential privacy can also be applied to ensure privacy preservation of data in the local updates [37], [85].

It is important to note that the FedAvg algorithm is a baseline algorithm presented in [139]. There exist many state-of-the-art algorithms with better performance than the FedAvg. For example, a new FL mechanism, FedOpt, was recently proposed in [138]. It was shown that the FedOpt is capable of achieving high accuracy, communication efficiency, and privacy preservation to outperform the FedAvg. Also, the HierFAVG mechanism in [146] presented improved communication efficiency to outperform the FedAvg. Similarly, the CE-FedAvg mechanism in [147] presented improved communication efficiency to outperform the FedAvg. We present the details of the FedOpt, HierFAVG, and CE-FedAvg mechanisms in later paragraphs.

Several benefits of using FL in edge caching systems were demonstrated in [37], [80], [85], [87]. We summarize the benefits as follows:

- **System becomes more cognitive** - In systems with a large number of UEs, the UEs can acquire various, abundant and personalized data for updating the global learning model. The data could include the quality of the wireless channel, the remaining battery life and the energy consumption, and the immediate computation capability. On the EDs, the cognitive data could include the computation load, the storage occupation, the number of wireless communication links, and the task queue states waiting for handling. As a result, the use of abundant and personalized distributed data instead of centralized training data ensures the system is more cognitive.
- **System becomes more robust** - Since FL can address the key issues such as the availability of the UEs, unbalanced data, and non-IID data, the performance of the systems is not easily affected by the unbalanced data or poor communication environment. Furthermore, its ability to handle non-IID data allows massive UEs in different wireless environments to train their own learning models without considering the overall negative effects.
- **Improved flexibility** - In FL, additional computation could be used to decrease the number of communication rounds required to train a model. The additional computation can be achieved by increasing the computation per UE. Therefore, UEs can decide to vary the number of mini-batches in training to adjust the communication cost.
- **Reduced network traffic and energy consumption** – The decentralized training can significantly reduce the network traffic and energy consumption by sending only the features of interest rather than the stream of raw data.
- **Stability despite loss of connectivity** - FL does not rely on synchronization among learners. Hence, even during a loss of connectivity between the EDs and UEs, the UEs can still build their local models. The feature is particularly important for highly dynamic and mission-critical applications [80].

Although the FL frameworks present promising results, the technique of FL has two main challenges. The technique suffers from the limitations of massive communication overhead and non-IID data [90], [148]. This is mainly because

the accuracy of FL algorithms is significantly reduced when highly non-IID data is used as compared to when IID data is used. The communication overhead of FL mainly comes from the global model aggregation and update [37], [138], [145]–[148]. For instance, there exists FL algorithms that require each UE to communicate its full gradient update during each epoch. Often, the update is the same size as the fully trained model, where the trained model could be in the size of gigabytes based on the learning architecture, and its millions of parameters. The size can increase to reach petabytes when the training is conducted on large-scale datasets that require thousands of training epochs [138]. The FedAvg is a good example of FL algorithms that suffer from massive communication overhead. In [142], it was shown that the communication overhead in FL can impact other parameters such as the model accuracy and training time.

For instance, when dynamic and heterogeneous large-scale IoT scenarios in [74], [85], [125], [142], [147] are considered, FL algorithms can be employed to enable the resource-constrained IoT devices to learn a shared learning model without centralizing the training data. However, the FL algorithms may achieve reduced accuracy compared to DRL algorithms, although the difference may be relatively negligible. As an example, the observations in [85], [125] show that it may not be feasible to perform fast training in FL. Thus, when FedAvg algorithm is used, learning takes a long time of training as well as inferring according to the required accuracy level. This is mainly because the FL algorithms require at least several effective model aggregations. In another scenario, in [37], it was shown that FL might involve heterogeneous devices with varying resource constraints. For instance, the devices can have different computing capabilities such as CPU states and battery level. Also, the devices can have different levels of willingness to participate. Considering the distributed nature of training across the numerous devices, it becomes possible to have free ridership [37]. Concerning the privacy preservation property of FL, it was shown in [37], [43], [140], [144] that some of the existing FL frameworks may still be susceptible to various privacy attacks.

In general, there are two main approaches to reduce communication overhead during the model training in FL:

- Reducing the total bits transferred for each UE update.
- Reducing the total number of updates transferred for each UE.

Therefore, [37], [90], [138], [145], [146]–[148] presented various mechanisms to address the challenge of massive communication overhead in FL. For instance, to address the challenges of the traditional FedAvg algorithm, novel FL frameworks were proposed in [138], [142], [146], [147]. The FedOpt framework in [138] was designed to address some challenges of the state-of-the-art FL frameworks. In particular, FedOpt addresses the challenges of massive communication overhead and privacy preservation. To ensure improved performance, FedOpt designs a new data compression algorithm and then integrates an efficient encryption technique with differential privacy. Similarly, the HierFAVG framework in [146] and CE-FedAvg framework in [147] achieved improved communication efficiency. The main idea of the CE-FedAvg was to update the FedAvg algorithm by reducing both the number of rounds taken to converge to a given accuracy and the total data uploaded during training over FedAvg. To ensure reduced communication overhead, CE-FedAvg employs a new compression technique. In the case of the HierFAVG framework, the main idea was to employ an intermediary structure in a hierarchical FL architecture so as to reduce the communication rounds between UEs and the servers. In particular, the HierFAVG algorithm allows multiple local aggregations at each ED before one global aggregation. Thus, the communication between the servers occurs only once after an interval of multiple updates. In contrast, the traditional FedAvg algorithm presented in [139] allows the global aggregation to occur more frequently since no intermediate edge server aggregation is involved.

A different strategy was considered in [145]. Instead of reducing the total bits transferred in each update via data compression, [145] presented an orthogonal approach that identifies irrelevant updates made by the UEs and precludes them from being uploaded. Therefore, [145] proposed the communication-mitigated federated learning (CMFL) algorithm that provides the UEs with the feedback information regarding the global tendency of model updating. Each UE checks if its update aligns with the global tendency and if it is relevant enough to model improvement. Therefore, by avoiding uploading the irrelevant updates to the server, CMFL reduces the communication overhead significantly while still guaranteeing the learning convergence.

In [148], [149], it was considered that it is more advantageous to employ analog transmission for global model aggregation than digital transmission. Furthermore, [148] presented that when digital transmission for global model aggregation was used, it resulted in increased communication latency as the number of UEs increased. Therefore, [148] presented an analog aggregation mechanism to reduce the communication overhead with respect to the number of UEs. Also, data redundancy was introduced to the system to deal with non-IID data. In [90], it was demonstrated that personalized federated learning can significantly reduce the performance degradations caused by the non-IID distribution. Therefore, [90] proposed a personalized federated learning framework to achieve high accuracy and reduced communication overhead. The framework considered lightweight models to ensure support for large-scale intelligent IoT applications. The work in [90] demonstrated the effectiveness of the FL frameworks through a case study of human activity recognition task. On the other hand, some interesting FL approaches were proposed in [150], [151]. The approach in [150] aimed to improve the communication efficiency and model performance for COVID-19 detection while [151] focused on utilizing FL mechanisms in a human mobility prediction framework to ensure good prediction performance and privacy protection of personal data on the UEs.

B. PERFORMANCE STUDY

The discussions in section V (A) show that the performance of the DL-based, DRL-based, and FL-based edge caching solutions can be affected by the type of learning algorithms used by the caching agent. For example, it is presented that the caching solutions present different performance when centralized or decentralized learning algorithms are used. It is discussed that decentralized algorithms present better performance than centralized algorithms. Therefore, in Table 3, we present a summary of the performance features of representative DL, DRL, and FL edge caching solutions. The solutions are adopted from [30], [74], [82], [84]–[86], [125], and [131]. Furthermore, we investigate the performance of the solutions. Our investigations are based on the analysis and evaluations performed in [30], [74], [82], [84]–[86], [125], [131].

The performance of the DL, DRL, and FL edge caching solutions is compared to the performance of two baseline solutions, the least recently used (LRU) and least frequently used (LFU) caching solutions which are commonly used by content providers [30], [86]. In the LRU algorithm, the system keeps track of the most recent requests for every cached content. When the cache storage becomes full, the cached content which is requested least recently, is replaced by the new content [30], [74]. For the LFU algorithm, the system keeps track of the number of requests for every cached content. When the cache storage becomes full, the cached content which is least frequently requested, is replaced by the new content [30], [74]. We compare the performance of the solutions in terms of content delivery latency and cache hit rate. The cache hit rate is used to show how frequently the requested content is found in the local cache [30], [86]. Therefore, we assume that cache hit rate indicates the content acquisition reliability. That means, a high cache hit rate corresponds to high content acquisition reliability.

It is shown in Table 3 that the DL, DRL, and FL edge caching solutions are capable of achieving improved performance in terms of content delivery latency and cache hit rate to outperform the LRU and LFU solutions. The main reason for the poor performance in the LRU and LFU solutions is that, the algorithms in LRU and LFU do not consider the popularity of contents in the future. As a result, the solutions do not adapt well to the dynamically changing content popularity and they achieve low cache efficiency [89], [113]. For instance, the LFU framework is not able to reach a good performance in IoT environment because it does not consider the saltation and timeliness of the IoT data popularity [113]. It was also shown in [74], [84], [86] that the DL, DRL, and FL edge caching solutions are capable of achieving reduced backhaul network traffic to outperform the LRU and LFU solutions.

On the other hand, Table 3 shows that although the solutions with centralized DL, DRL, and FL algorithms perform better than the LRU and LFU solutions, the centralized solutions present reduced performance when compared to the solutions with decentralized algorithms. For example, it was shown in [82] that a DDL solution performs better than a centralized DL solution in terms of latency and cache hit rate. Furthermore, since the DDL framework only needs to collect the trained models from the EDs without considering any raw dataset transmission, the DDL framework was able to achieve reduced communication overhead. Also, the DDL was able to learn the dataset faster than the centralized DL as the number of the EDs was increased. It was also presented in [30], [86] that decentralized MADRL frameworks achieve better performance than centralized DRL frameworks. As an example, when a massively vibrant, diversified, and distributed video streaming environment was considered in [86], a decentralized MADRL framework presented better performance than a centralized DRL framework in terms of video access latency and the traffic cost. In [85], it was revealed that a FL scheme can consume significantly lower communication resources than a centralized DRL framework. In [85], [125], it was presented that the FL-based solutions achieve a lower number of dropped tasks, queuing delay, and transmission energy to outperform the DRL solutions.

Conversely, when the performance of the FL frameworks was investigated in [74], [85], [125], it was revealed that although FL schemes can address the challenges of DRL and DL frameworks, the FL schemes are less capable of achieving significantly improved performance in terms of content access latency and cache hit rate. For example, the performance of the FL schemes in [74], [85], [125] was comparable to the performance of the centralized DRL schemes once the model aggregation of FL was performed several times. That means, for the FL algorithms to achieve the performance level of the centralized DRL algorithms in terms of content access latency and cache hit rate, the FL algorithms must allow several rounds of model aggregation. On the other hand, it was pointed out in [74], [125] that the performance level of the FL schemes is reasonable since the FL schemes assume more practical network conditions. As an example, the centralized DRL scheme considered in [74], [85], [125] assumed that the massive training data can be successfully uploaded to the ED without loss or delay. Considering the limitations of wireless channels, the assumption made by the DRL scheme may be impractical. Moreover, the work in [37], [85] considered the challenge that when non-IID datasets are used, FL algorithms incur large number of communication rounds to converge to the global optimal. Consequently, [37], [85], [90] highlighted the technique of transfer learning as a potential solution for the challenge. Thus, [37], [85], [90] considered the use of transfer learning technique to improve the learning efficiency of the FL algorithms. It was pointed out in [37] that the transfer learning technique can ensure training is not initialized from scratch. In [90], it was demonstrated that personalized federated learning can significantly reduce the performance degradations caused by the non-IID data.

VI. SUMMARY AND DISCUSSIONS

There exists many solutions to guarantee low latency and reliable communications in 5G and beyond networks. We have discussed various techniques and solutions which are based on the mechanisms of configurable subcarrier spacing, grant-free access, edge computing, edge caching, in-network caching, network slicing, dynamic multiplexing, latency sensitive scheduling schemes, TSN, and DetNet.

It is considered that 5G and beyond networks will assume decentralized and infrastructureless communication to enable devices to cooperate directly over D2D spontaneous connections. The networks are designed to operate when needed without the support of a central coordinator or with limited support for synchronization and signaling. Thus, the networks assume content servers and UEs that manage part of the computing tasks locally to ensure fast feedback and low latency [136]. Hence, 5G and beyond networks will experience a shift from the conventional cloud-based computing setting to edge computing systems. Different from the cloud-based computing systems which aggregate the computational resources in data centers, the edge computing systems deploy computational power at the network edges. The shift is beneficial for applications that demand URLLC, as well as supporting resource-constrained nodes reachable only over unreliable network connections. Together with the development of AI-enabled technologies, future networks will be able to utilize local data to conduct intelligent inference and control on many activities to ensure low latency and reliable communications in various application scenarios, including 5G video streaming services, smart cities, social IoT, intelligent transport systems, and IIoT.

It is observed that FL is a viable mechanism in 5G and beyond networks. FL edge caching frameworks guarantee many advantageous performance features for future intelligent networks. The frameworks are capable of addressing the challenges of the conventional centralized and decentralized edge caching schemes. In particular, the FL schemes present significant performance gains over DL and DRL schemes. It is shown that the advantages of using FL edge caching include ability to handle non-IID distributed and unbalanced data, the systems become more cognitive and robust, improved system flexibility, availability of larger data for training, reduced network traffic and energy consumption, and privacy preservation of training data. Thus, FL frameworks are highly feasible in mission-critical and privacy-sensitive applications such as e-healthcare, remote education, and urban vehicular networks. However, traditional FL algorithms incur high communication overhead due to the process of global model aggregation and update. Therefore, recent studies are investigating the approaches for achieveing reduced communication overhead in FL schemes.

Furthermore, it is shown that B5G networks will focus on machine-based vertical applications such as autonomous driving, unmanned aerial vehicle services, autonomous services, robotic applications, and IIoT. These applications require deterministic and bounded ULL. Furthermore, in some cases, high data rate is required for transporting video feeds from cameras that are used to control vehicles and robots. In application scenarios such as autonomous driving and industrial automation and control systems, the required data rates and bounded ULL is achived by employing the mechanisms of TSN and DetNet standards.

Therefore, considering the popularity of intelligent transport systems and industrial automation and control systems in 5G and beyond networks, we assume that the enabling techniques which are based on mobile edge AI, FL, TSN, and DetNet will play important roles in 5G and beyond networks, to bring numerous benefits and boost the economy. For example, with mobile edge intelligence and FL it may be possible to devise future intelligent Internet of vehicles which is an emerging paradigm for the automobile industry. Automakers and their suppliers are responsible for 3% of the United States gross domestic product (GDP), and no other manufacturing sector generates as many jobs in the United States [28]. In 2018, the United States and China, the two largest automobile markets, sold 17.2 and 23.2 million passenger cars, respectively [28]. Moreover, automotive manufacturers and technology companies, such as Tesla, Google, Mobileye, and Uber, are investing heavily in intelligent vehicles, with the 2018 Audi A8 being the first self-driving car available in production [28].

VII. OPEN ISSUES AND OPPORTUNITIES

In this section, we outline the technical challenges and open issues in the mechanisms of configurable subcarrier spacing and sTTI, grant-free access, scheduling and multiplexing schemes, network slicing, edge caching, in-network caching, TSN, and DetNet.

A. CONFIGURABLE SUBCARRIER SPACING AND sTTI

To ensure URLLC, several challenges and system design tradeoffs are incurred. The tradeoffs may be in the form of sTTI versus control overhead, spectral efficiency versus latency, reliability versus latency and rate, device energy consumption versus latency, or energy consumption versus reliability [1], [45], [52]. Therefore, the tradeoffs must be considered during the system designing. For example, to achieve low latency, spectral efficiency penalty may be incurred due to HARQ and sTTI. Thus, a system characterization of spectral efficiency versus latency in a multiple access and broadcast system must consider issues such as bursty packet arrivals and a mixture of low latency and delay tolerant traffic [45]. In another example, fewer participating edge caching nodes can provide better reliability, but it can also result in increased latency. Therefore, there is an opportunity for further research to address the issues concerning the tradeoffs between latency and reliability. Some of the tradeoff issues were considered in [45].

B. SCHEDULING SCHEMES AND NETWORK SLICING

When the coexistence of eMBB and URLLC traffic must be ensured, the queueing effect may have a significant impact on the performance of URLLC because of the hard latency requirements [17]. This is mainly due to the different scheduling granularity. These issues can benefit from further research. Furthermore, there are many issues that need to be addressed before the practical implementation of network slicing becomes a reality [66]. For instance, more research is required to devise flexible network slices allocation policies which are able to adapt their behavior to the customer requests. The analysis of customer behavior is one of the hardest challenges during the prediction of customer requests [66]. In some scenarios, it is important to consider the privacy issues related to collection of sensitive user data. In such scenarios, FL frameworks may be considered [66]. In [133], it was highlighted that a major network slicing requirement is traffic isolation and security enforcement. This is mainly because each network slice should not be able to access the traffic or other information of other slices/services. Therefore, there is an opportunity for further research on the issues of communication efficiency, traffic isolation, and security. Some of the issues were considered in [133], [152]. On the other hand, it is observed that more studies are required to devise FL frameworks for network slicing. For example, a recent work in [66] was the first study to employ FL mechanisms for network slicing. Therefore, future research may focus on exploiting the benefits of FL algorithms for network slicing, especially in privacy-sensitive applications.

C. GRANT-FREE ACCESS

It was shown in [53] that in grant-free access techniques, the mechanism of data replication might be ineffective and it cannot support URLLC for services with sporadic traffic. This is mainly due to the impact of self-collisions. Therefore, it is recommended that new approaches should be considered to minimize collisions between UEs sharing radio resources. However, the new approaches cannot be based either on grant-based or semi-persistent scheduling. This is mainly due to the fact that grant-based scheduling introduces additional latency due to the exchange of messages between the UEs and the base station for assigning the radio resources. Furthermore, semi-persistent scheduling with dedicated resources inefficiently utilizes the available resources when considering dedicated resources and sporadic traffic. Future research should devise new grant-free scheduling approaches by considering the use of sensing mechanisms or full duplex techniques that can reduce packet collisions [53].

D. EDGE CACHING AND IN-NETWORK CACHING

Although many studies have been done to address the issues of edge caching and in-network caching, many issues remain open for more research. For example, the work in [34], [89], [113], [120], [131] considered the future popularity of content while making cache decisions. In [113], various fundamental questions about IoT data popularity and related popularity-based caching were considered. However, it was highlighted in [120] that in IoT devices with limited caching resources, caching the most popular content may not always be the optimal strategy. Instead, in such scenarios, probabilistic caching strategies may be considered where the contents are independently placed according to their caching probabilities. Therefore, there is an opportunity for more exploration on the idea of content popularity and probabilistic caching strategies for making the cache decisions. Furthermore, it was pointed out in [85] that general DL-based optimization and prediction schemes take quite a long running time of recursions for converging to the results, which is inappropriate for URLLC traffic. To achieve the required performance, caching solutions should be able to provide differentiated support for various types of services, and fine-grained collaborative scheduling of the AI tasks over the edge nodes and mobile devices should be accelerated in nearly real-time. In addition, it was discussed in [9] that edge computing techniques can be integrated with various other techniques to solve several challenges of 5G and B5G networks. For example, it was highlighted that the 5G and B5G networks can benefit from the integration of edge computing techniques with other technologies such as NOMA, mmW, and mMIMO. In particular, the coexistence of edge computing techniques with mmW and mMIMO is necessary to enable massive wireless connectivity with high data rates, low latency, and large computing capabilities. Thus, future research can focus on investigating the methods for achieving the integrations.

New solutions which are based on FL frameworks are effective at addressing multiple challenges, as shown in [37], [66], [87], [88]. With FL algorithms, it is possible to address the latency and reliability issues together with energy, bandwidth, data privacy, and security concerns. Despite the apparent prospects, there are many open issues for further research. For example, although the existing FL frameworks present promising results when used in edge caching systems, it was shown that for FL frameworks to achieve a target accuracy, they require several communication rounds for model aggregation and update. Thus, there is a need for further research to improve the communication efficiency of FL. Some of the techniques for improving the communication efficiency of FL were highlighted in [37], [85], [138]. In [148], [149] it was considered that it is more advantageous to employ analog transmission for global model aggregation than digital transmission. Therefore, future research may focus on investigating the techniques for analog transmissions where the global model is broadcasted by the remote server in an uncoded manner. On the other hand, it was presented in [85] that more research can be done to investigate how the technique of transfer learning can be used to reduce the learning time in FL algorithms. The transfer learning techniques can be used to increase the efficiency of training by ensuring training is not initialized from scratch [37].

The use of transfer learning technique was also discussed in [37], [41], [90], [141].

In [90], it was discussed that personalized federated learning approaches can be used to address the device heterogeneity challenges of FL in complex IoT environments. Thus, it is important to perform personalization in device, data, and model levels so as to mitigate heterogeneities and attain high-quality personalized model for each device. Consequently, future research should investigate the operational features of the emerging personalized federated learning approaches such as federated transfer learning, federated meta learning, federated multi-task learning, and federated distillation. Moreover, more research is needed to improve the performance of FL frameworks as it was shown in [39], [43], [140] that although the frameworks are effective at reducing latency, they may be less effective at guaranteeing the required data privacy. Several issues regarding FL techniques were discussed in the recent work of [37], [39], [40], [90], [138], [142], [144], [146], [147], [152]–[154]. The work in [154] discussed the technical challenges of applying FL in vehicular IoT, and highlighted the necessary improvements that should be made for IoT technologies. It highlighted that in order to achieve a seamless integration of FL and vehicular IoT applications, it is important to do further investigations and design vehicular environment-aware FL protocols and algorithms.

E. ON-DEVICE ML AND AI

It can be challenging to ensure URLLC in IoT devices through the use of on-device AI. This is mainly due to the fact that most of the IoT devices have limited resources including limited storage and computational capabilities. Hence, the devices may not be able to carry out AI-based or machine learning algorithms effectively while meeting the reliability and latency requirements [4], [85]. Therefore, there is an opportunity for more research to devise DL algorithms which are better suited for the resource-constrained IoT devices. As an example, the study in [130] presented DL algorithms which are more suitable for the IoT devices. In [136], [142], FL algorithms were presented for resource-constrained IoT devices. Also, the challenges for deploying traditional FL algorithms in complex IoT environments were considered in [90].

The recent work in [8] presented a different perspective for URLLC. It stated that information latency might be more relevant than communication latency for URLLC in wireless networked control (WNC) systems such as autonomous driving and the IIoT. For instance, in wireless networked factory automation, several control centers collect status information from distributed terminals through a wireless network and then disseminate the control signals to actuators that carry out the actions. Control is based on the perceived information and status of the physical world and the virtual world. Therefore, [8] defined information latency between the virtual and physical worlds as a rational measure of the time periods elapsed since the last true statuses of the physical world are observed and input to the virtual world. Thus, information latency is directly related to the information timeliness observed at the control decision maker. When the physical world is completely synchronized with the statuses of the virtual world, the information latency becomes zero, URLLC is achieved, and the performance of WNC systems is optimized. Therefore, there is an opportunity for more research to devise solutions to optimize information latency in WNC systems through the use of AI. A prominent research direction may be to devise a novel network architecture that supports AI-assisted, situationally-aware wireless systems. A situationally-aware multi-agent RL framework was proposed in [8] to realize optimized information latency in autonomous driving scenarios. It was shown that good system performance can be achieved through information latency optimized control.

F. TSN AND DETNET STANDARDS

The TSN and DetNet standards present some promising mechanisms in traffic flows with deterministic ULL. The standards ensure traffic flow adapts to the desired delay bound and receives immediate and highly robust delivery. However, several issues remain open for more research as shown in [16], [96], [98], [155]–[159]. For instance, it was presented in [16] that reliable, secure, and low-latency communication between multiple TSN networks is essential to support a wide range of future applications spanning from time-sensitive to delay tolerant applications with flow level scheduling capabilities. Thus, it is important to explore about the TSN standards for connecting and communicating with external TSN and non-TSN networks specifically for interoperating networks. In the case of the IEEE 802.1 CM standard, it is observed that the delay and synchronization aspects have been specified in the standard. However, more research is required to specify the issues of security and reliability. Moreover, the deterministic TSN network behavior has so far been generally applied to a closed network, covering only the scope of in-vehicle networks and industrial automation. Nevertheless, the connectivity to external networks, such as cellular and WLAN networks will enhance the capabilities of the TSN networks. Therefore, future research may focus on finding the mechanisms to ensure connectivity between multiple closed TSN architectures.

The work in [156], [159] discussed the open issues regarding the mechanisms for the integration of TSN and 5G while [158] focused on highlighting the open issues for investigations to analyze the impact of conveying time-sensitive and best-effort traffic on a common metro infrastructure. It was observed that in order to achieve seamless integration of 5G and TSN, it is necessary to do further investigations on the configuration and scheduling mechanisms. This is mainly because the support of TSN functionalities in 5G alone are not sufficient to provide the necessary performance required by TSN. It was shown that depending on the application configuration and inter-arrival time of frames belonging to different streams, providable service guarantees on reliability and

latency vary widely. Thus, in order to achieve a feasible integration of 5G into TSN, it is important for future research to consider the effects when devising an end-to-end scheduling, in order to assign resources on the shared wired and wireless link to the respective streams efficiently. Moreover, the work in [155] presented an overview of ongoing standardization activities in 3GPP on the topic of integration of 5G and TSN for future industrial communication infrastructures. Also, the work discussed several issues and open topics related to the requirements of an integrated industrial network.

The work in [96] considered the issue of real-time scheduling of massive data in TSN with a limited number of schedule entries. It pointed out that in industrial applications, the backbone network connects multiple industrial field networks together and has to carry massive real-time packets. However, the off-the-shelf TSN switches can deterministically schedule no more than 1024 real-time flows due to the limited number of schedule table entries. The excess real-time flows have to be delivered by best-effort services because the switch only supports the two scheduling services. The best-effort services can reduce average delay, but cannot guarantee the hard real-time constraints of industrial applications. Therefore, future research can focus on exploring the approaches to make a limited number of schedule table entries support more real-time flows. In [157] it was discussed that the use of one-level frame preemption paradigm in IEEE 802.1Qbu standard can result in a challenge that some preemptable frames can still suffer long blocking periods, irrespective of their individual priority levels. Therefore, to address the challenge, [157] proposed the use of multi-level preemption paradigm. Thus, based on the findings of [157], future work should investigate the use of multi-level preemption techniques for the IEEE 802.1Qbu standard. The use of semi-persistent scheduling mechanisms for TSN was considered in [11], [159]. In [159], it was presented that the use of semi-persistent scheduling in the downlink or configured scheduling with periodic resources can ensure support for periodic traffic. However, more research is required to analyze the issues concerning traffic periodicity when the scheduling mechnisms are employed.

On the other hand, it was highlighted in [16] that although DetNet focuses on the network layer (L3) and higher layers, DetNet relies on the time sensitive link layer to establish the deterministic L3 packet flow properties. Hence, it is important to devise DetNet mechanisms which are independent of the time sensitive link layer to allow wide adoption of DetNet. Therefore, future research should consider investigating various techniques for improving the practicality of the TSN and DetNet protocols while promoting DetNet mechanisms which are independent of the time sensitive link layer.

VIII. CONCLUSION

The mission-critical applications in 5G and beyond networks require extremely low delays on the order of 1 ms with very high reliability of about 99.999%. In some applications, ultra-low latency of below 1 ms is required. For example, the end-to-end latencies for industrial automation should be on the order of a few μs to a few ms, around 1 ms or below for the Tactile Internet, and on the order of 100 μs for the one-way fronthaul in wireless cellular networks. Therefore, this article presents an overview of the enabling techniques for low latency and reliable communications in 5G and beyond networks. A classification of the enabling techniques is presented, highlighting the performance features of various techniques. In particular, we discuss the mechanisms of dynamic multiplexing, grant-free access, latency sensitive scheduling, network slicing, sTTI, edge computing, edge caching, in-network caching, TSN, and DetNet. Subsequently, a performance study of DL, DRL, and FL edge caching schemes is presented. We investigate the performance of the schemes in various application scenarios while considering the settings of centralized and decentralized edge caching systems.

It is shown that FL edge caching frameworks are capable of training a globally shared model by exploiting a massive amount of user-generated data samples on user equipment while preventing data leakage. This is achieved by exchanging only model parameters learned locally at each user equipment. As a result, FL limits the amount of data to be transferred and reduces the impact of data breaches. Thus, FL edge caching ensures reduced content delivery latency, improved content acquisition reliability, and reduced network traffic and energy consumption while guaranteeing the use of training data from user equipment without sacrificing the personal data privacy. Furthermore, FL edge caching presents effective mechanisms for dynamic and heterogeneous large-scale networks where devices are resource-constrained, including in complex IoT environments. Therefore, FL becomes a viable mechanism in 5G and beyond networks. However, traditional FL algorithms incur high communication overhead. Hence, recent studies are devising new FL algorithms to improve the communication efficiency.

On the other hand, the TSN and DetNet standards present effective mechanisms when deterministic networking and bounded ultra-low latency are considered. Specifically, the TSN and DetNet standards emerge as practical approaches in applications such as automotive in-vehicle networking and industrial automation and control systems. Finally, the technical challenges and open issues are presented for further research.

REFERENCES

[1] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3098–3130, 2018.

[2] T. Fehrenbach, R. Datta, B. Goktepe, T. Wirth, and C. Hellge, "URLLC services in 5G low latency enhancements for LTE," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–6.

[3] S. K. Sharma, I. Woungang, A. Anpalagan, and S. Chatzinotas, "Toward tactile Internet in beyond 5G era: Recent advances, current issues, and future directions," *IEEE Access*, vol. 8, pp. 56948–56991, 2020.

[4] M. A. Siddiqi, H. Yu, and J. Joung, "5G ultra-reliable low-latency communication implementation challenges and operational issues with IoT devices," *Electronics*, vol. 8, no. 9, p. 981, Sep. 2019.

[5] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55765–55779, 2018.

[6] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 124–130, Jun. 2018.

[7] G. Pocovi, H. Shariatmadari, G. Berardinelli, K. Pedersen, J. Steiner, and Z. Li, "Achieving ultra-reliable low-latency communications: Challenges and envisioned system enhancements," *IEEE Netw.*, vol. 32, no. 2, pp. 8–15, Mar. 2018.

[8] Z. Jiang, S. Fu, S. Zhou, Z. Niu, S. Zhang, and S. Xu, "AI-assisted low information latency wireless networking," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 108–115, Feb. 2020.

[9] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, Jun. 2020.

[10] K. Samdanis and T. Taleb, "The road beyond 5G: A vision and insight of the key technologies," *IEEE Netw.*, vol. 34, no. 2, pp. 135–141, Mar. 2020.

[11] R. B. Abreu, G. Pocovi, T. H. Jacobsen, M. Centenaro, K. I. Pedersen, and T. E. Kolding, "Scheduling enhancements and performance evaluation of downlink 5G time-sensitive communications," *IEEE Access*, vol. 8, pp. 128106–128115, Jul. 2020.

[12] M. Iwabuchi, A. Benjebbour, Y. Kishiyama, G. Ren, C. Tang, T. Tian, L. Gu, T. Takada, and T. Kashima, "5G field experimental trials on URLLC using new frame structure," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2017, pp. 1–6.

[13] A. Anand, G. de Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 477–490, Apr. 2020.

[14] Y. Xu, C. Shen, T.-H. Chang, S.-C. Lin, Y. Zhao, and G. Zhu, "Transmission energy minimization for heterogeneous low-latency NOMA downlink," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1054–1069, Feb. 2020.

[15] H. Shariatmadari, S. Iraji, R. Jantti, P. Popovski, Z. Li, and M. A. Uusitalo, "Fifth-generation control channel design: Achieving ultrareliable low-latency communications," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 84–93, Jun. 2018.

[16] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 88–145, 1st Quart., 2019.

[17] C.-P. Li, J. Jiang, W. Chen, T. Ji, and J. Smee, "5G ultra-reliable and low-latency systems design," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2017, pp. 1–5.

[18] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, vol. 6, pp. 12825–12837, 2018.

[19] J. Sachs, G. Wikström, T. Dudda, R. Baldemair, and K. Kittichokechai, "5G radio network design for ultra-reliable low-latency communication," *IEEE Netw.*, vol. 32, no. 2, pp. 24–31, Mar. 2018.

[20] M. A. Lema, A. Laya, T. Mahmoodi, M. Cuevas, J. Sachs, J. Markendahl, and M. Dohler, "Business case and technology analysis for 5G low latency applications," *IEEE Access*, vol. 5, pp. 5917–5935, 2017.

[21] J. Zeng, T. Lv, Z. Lin, R. P. Liu, J. Mei, W. Ni, and Y. J. Guo, "Achieving ultrareliable and low-latency communications in IoT by FD-SCMA," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 363–378, Jan. 2020.

[22] W. Yang, C.-P. Li, A. Fakoorian, K. Hosseini, and W. Chen, "Dynamic URLLC and eMBB multiplexing design in 5G new radio," in *Proc. IEEE 17th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2020, pp. 1–5.

[23] L. Feng, Y. Zi, W. Li, F. Zhou, P. Yu, and M. Kadoch, "Dynamic resource allocation with RAN slicing and scheduling for uRLLC and eMBB hybrid services," *IEEE Access*, vol. 8, pp. 34538–34551, Feb. 2020.

[24] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[25] L. Zhao, J. Wang, J. Liu, and N. Kato, "Optimal edge resource allocation in IoT-based smart cities," *IEEE Netw.*, vol. 33, no. 2, pp. 30–35, Mar. 2019.

[26] H. Asmat, I. U. Din, F. Ullah, M. Talha, M. Khan, and M. Guizani, "ELC: Edge linked caching for content updating in information-centric Internet of Things," *Comput. Commun.*, vol. 156, pp. 174–182, Apr. 2020.

[27] C. Gong, F. Lin, X. Gong, and Y. Lu, "Intelligent cooperative edge computing in the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9372–9382, Oct. 2020.

[28] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the Internet of vehicles," *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, Feb. 2020.

[29] H.-T. Chien, Y.-D. Lin, C.-L. Lai, and C.-T. Wang, "End-to-end slicing with optimized communication and computing resource allocation in multi-tenant 5G systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2079–2091, Feb. 2020.

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

[31] L. Lo Bello and W. Steiner, "A perspective on IEEE time-sensitive networking for industrial communication and automation systems," *Proc. IEEE*, vol. 107, no. 6, pp. 1094–1120, Jun. 2019.

[32] J. Prados-Garzon, T. Taleb, and M. Bagaa, "LEARNET: Reinforcement learning based flow scheduling for asynchronous deterministic networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 7–11.

[33] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Routing and scheduling of time-triggered traffic in time-sensitive networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 452–4534, Jul. 2020.

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

[35] S. Arshad, M. A. Azam, M. H. Rehmani, and J. Loo, "Recent advances in information-centric networking based Internet of Things (ICN-IoT)," *IEEE Internet Of Things J.*, vol. 14, no. 8, pp. 1–31, Sep. 2018.

[36] J. Pfender, A. Valera, and W. K. G. Seah, "Content delivery latency of caching strategies for information-centric IoT," 2019, *arXiv:1905.01011*. [Online]. Available: http://arxiv.org/abs/1905.01011

[37] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

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

[39] B. Brik, A. Ksentini, and M. Bouaziz, "Federated learning for UAVs-enabled wireless networks: Use cases, challenges, and open problems," *IEEE Access*, vol. 8, pp. 53841–53849, Mar. 2020.

[40] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[41] J. Shuja, K. Bilal, W. Alasmary, H. Sinky, and E. Alanazi, "Applying machine learning techniques for caching in edge networks: A comprehensive survey," Jun. 2020, *arXiv:2006.16864*. [Online]. Available: http://arxiv.org/abs/2006.16864

[42] M. Mehrabi, D. You, V. Latzko, H. Salah, M. Reisslein, and F. H. P. Fitzek, "Device-enhanced MEC: Multi-access edge computing (MEC) aided by end device computation and caching: A survey," *IEEE Access*, vol. 7, pp. 166079–166108, Nov. 2019.

[43] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 19–23.

[44] A. Nasrallah, V. Balasubramanian, A. Thyagaturu, M. Reisslein, and H. ElBakoury, "TSN algorithms for large scale networks: A survey and conceptual comparison," 2019, *arXiv:1905.08478*. [Online]. Available: http://arxiv.org/abs/1905.08478

[45] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.

[46] Y. Hu, M. C. Gursoy, and A. Schmeink, "Relaying-enabled ultra-reliable low-latency communications in 5G," *IEEE Netw.*, vol. 32, no. 2, pp. 62–68, Mar. 2018.

[47] V. W. S. Wong, R. Schober, D. W. K. Ng, and L.-C. Wang, Eds., *Key Technologies for 5G Wireless Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[48] Z. Ma, M. Xiao, Y. Xiao, Z. Pang, H. V. Poor, and B. Vucetic, "High-reliability and low-latency wireless communication for Internet of Things: Challenges, fundamentals, and enabling technologies," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7946–7970, Oct. 2019.

[49] G. J. Sutton, J. Zeng, R. P. Liu, W. Ni, D. N. Nguyen, B. A. Jayawickrama, X. Huang, M. Abolhasan, and Z. Zhang, "Enabling ultra-reliable and low-latency communications through unlicensed spectrum," *IEEE Netw.*, vol. 32, no. 2, pp. 70–77, Mar. 2018.

[50] G. Hampel, C. Li, and J. Li, "5G ultra-reliable low-latency communications in factory automation leveraging licensed and unlicensed bands," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 117–123, May 2019.

[51] M. Khoshnevisan, V. Joseph, P. Gupta, F. Meshkati, R. Prakash, and P. Tinnakornsrisuphap, "5G industrial networks with CoMP for URLLC and time sensitive network architecture," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 947–959, Apr. 2019.

[52] G. Pocovi, K. I. Pedersen, and P. Mogensen, "Joint link adaptation and scheduling for 5G ultra-reliable low-latency communications," *IEEE Access*, vol. 6, pp. 28912–28922, May 2018.

[53] M. C. Lucas-Estañ, J. Gozalvez, and M. Sepulcre, "On the capacity of 5G NR grant-free scheduling with shared radio resources to support ultra-reliable and low-latency communications," *Sensors*, vol. 19, no. 16, p. 3575, Aug. 2019.

[54] Y. Chen, A. Bayesteh, Y. Wu, B. Ren, S. Kang, S. Sun, Q. Xiong, C. Qian, B. Yu, Z. Ding, S. Wang, S. Han, X. Hou, H. Lin, R. Visoz, and R. Razavi, "Toward the standardization of non-orthogonal multiple access for next generation wireless networks," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 19–27, Mar. 2018.

[55] X. Chen, D. W. K. Ng, W. Yu, E. G. Larsson, N. Al-Dhahir, and R. Schober, "Massive access for 5G and beyond," *IEEE J. Sel. Areas Commun.*, to be published.

[56] C. Wang, Y. Chen, Y. Wu, and L. Zhang, "Performance evaluation of grant-free transmission for uplink URLLC services," in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, Jun. 2017, pp. 1–6.

[57] N. H. Mahmood, R. Abreu, R. Bohnke, M. Schubert, G. Berardinelli, and T. H. Jacobsen, "Uplink grant-free access solutions for URLLC services in 5G new radio," in *Proc. 16th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2019, pp. 607–612.

[58] H. Mu, Z. Ma, M. Alhaji, P. Fan, and D. Chen, "A fixed low complexity message pass algorithm detector for up-link SCMA system," *IEEE Wireless Commun. Lett.*, vol. 4, no. 6, pp. 585–588, Dec. 2015.

[59] Y. Wu, S. Zhang, and Y. Chen, "Iterative multiuser receiver in sparse code multiple access systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 2918–2923.

[60] L. Yang, Y. Liu, and Y. Siu, "Low complexity message passing algorithm for SCMA system," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2466–2469, Dec. 2016.

[61] F. Wei and W. Chen, "Low complexity iterative receiver design for sparse code multiple access," *IEEE Trans. Commun.*, vol. 65, no. 2, pp. 621–634, Feb. 2017.

[62] L. Yang, X. Ma, and Y. Siu, "Low complexity MPA detector based on sphere decoding for SCMA," *IEEE Commun. Lett.*, vol. 21, no. 8, pp. 1855–1858, Aug. 2017.

[63] C. Zhang, Y. Luo, and Y. Chen, "A low-complexity SCMA detector based on discretization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2333–2345, Apr. 2018.

[64] H.-T. Chien, Y.-D. Lin, C.-L. Lai, and C.-T. Wang, "End-to-end slicing as a service with computing and communication resource allocation for multi-tenant 5G systems," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 104–112, Oct. 2019.

[65] M. Alsenwi, N. H. Tran, M. Bennis, A. K. Bairagi, and C. S. Hong, "EMBB-URLLC resource slicing: A risk-sensitive approach," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 740–743, Apr. 2019.

[66] R. Fantacci and B. Picano, "When network slicing meets prospect theory: A service provider revenue maximization framework," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3179–3189, Mar. 2020.

[67] S. O. Oladejo and O. E. Falowo, "Latency-aware dynamic resource allocation scheme for multi-tier 5G network: A network slicing-multitenancy scenario," *IEEE Access*, vol. 8, pp. 74834–74852, Apr. 2020.

[68] A. Matera, R. Kassab, O. Simeone, and U. Spagnolini, "Non-orthogonal eMBB-URLLC radio access for cloud radio access networks with analog fronthauling," *Entropy*, vol. 20, no. 9, p. 661, Sep. 2018.

[69] L. Zhang, J. Liu, M. Xiao, G. Wu, Y.-C. Liang, and S. Li, "Performance analysis and optimization in downlink NOMA systems with cooperative full-duplex relaying," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 10, pp. 2398–2412, Oct. 2017.

[70] Z. Wei, L. Yang, D. W. K. Ng, J. Yuan, and L. Hanzo, "On the performance gain of NOMA over OMA in uplink communication systems," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 536–568, Jan. 2020.

[71] S. Schiessl, M. Skoglund, and J. Gross, "NOMA in the uplink: Delay analysis with imperfect CSI and finite-length coding," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 3879–3893, Jun. 2020.

[72] J. García-morales, M. C. Lucas-Estañ, and J. Gozalvez, "Latency-sensitive 5G RAN slicing for industry 4.0," *IEEE Access*, vol. 7, pp. 143139–143159, Sep. 2019.

[73] X. Li, X. Wang, P.-J. Wan, Z. Han, and V. C. M. Leung, "Hierarchical edge caching in Device-to-Device aided mobile networks: Modeling, optimization, and design," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1768–1785, Aug. 2018.

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

[75] S. He, J. Ren, J. Wang, Y. Huang, Y. Zhang, W. Zhuang, and S. Shen, "Cloud-edge coordinated processing: low-latency multicasting transmission," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1144–1158, May 2019.

[76] T. Zhang, X. Fang, Y. Liu, and A. Nallanathan, "Content-centric mobile edge caching," *IEEE Access*, vol. 8, pp. 11722–11731, Jan. 2020.

[77] U. Paul, J. Liu, S. Troia, O. Falowo, and G. Maier, "Traffic-profile and machine learning based regional data center design and operation for 5G network," *J. Commun. Netw.*, vol. 21, no. 6, pp. 569–583, Dec. 2019.

[78] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, "Cooperative edge caching in user-centric clustered mobile networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1791–1805, Aug. 2018.

[79] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2020.

[80] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Federated learning for ultra-reliable low-latency V2 V communications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[81] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1610.02527

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

[83] X. Fan, Y. Huang, X. Ma, J. Liu, and V. C. M. Leung, "Exploiting the edge power: An edge deep learning framework," *CCF Trans. Netw.*, vol. 2, no. 1, pp. 4–11, Dec. 2018.

[84] H. Pang, J. Liu, X. Fan, and L. Sun, "Toward smart and cooperative edge caching for 5G networks: A deep learning based approach," in *Proc. IWQoS*, Jun. 2018, pp. 1–6.

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

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

[87] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities and challenges," 2019, *arXiv:1908.06847*. [Online]. Available: http://arxiv.org/abs/1908.06847

[88] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2134–2143, Mar. 2020.

[89] Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas, "Federated learning based proactive content caching in edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[90] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent IoT applications: A cloud-edge based framework," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 35–44, May 2020.

[91] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[92] X. Yang, D. Scholz, and M. Helm. *Deterministic Networking (DetNet) vs Time Sensitive Networking (TSN)*. Accessed: Oct. 2020. [Online]. Available: https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2019-10-1.pdf

[93] Y. Li, P. Zhang, Y. Zhou, and D. Jin, "A data forwarding mechanism based on deep reinforcement learning for deterministic networks," in *Proc. INFOCOM IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Jul. 2020, pp. 6–9.

[94] V. Addanki and L. Iannone, "Moving a step forward in the quest for deterministic networks (DetNet)," in *Proc. IFIP Netw. Conf. (Netw.)*, Jun. 2020, pp. 22–26.

[95] S. Bhattacharjee, R. Schmidt, K. Katsalis, C.-Y. Chang, T. Bauschert, and N. Nikaein, "Time-sensitive networking for 5G fronthaul networks," in *Proc. ICC*, Jun. 2020, pp. 7–11.

[96] X. Jin, C. Xia, N. Guan, C. Xu, D. Li, Y. Yin, and P. Zeng, "Real-time scheduling of massive data in time sensitive networks with a limited number of schedule entries," *IEEE Access*, vol. 8, pp. 6751–6767, Jan. 2020.

[97] N. Desai and S. Punnekkat, "Enhancing fault detection in time sensitive networks using machine learning," in *Proc. COMSNETS*, Jan. 2020, pp. 7–11.

[98] A. Badar, D. Z. Lou, U. Graf, C. Barth, and C. Stich, "Intelligent edge control with deterministic-IP based industrial communication in process automation," in *Proc. 15th CNSM*, Oct. 2019, pp. 21–25.

[99] J. Zeng, T. Lv, R. P. Liu, X. Su, Y. J. Guo, and N. C. Beaulieu, "Enabling ultrareliable and low-latency communications under shadow fading by massive MU-MIMO," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 234–246, Jan. 2020.

[100] Z. Wang, T. Lv, Z. Lin, J. Zeng, and P. T. Mathiopoulos, "Outage performance of URLLC NOMA systems with wireless power transfer," *IEEE Wireless Commun. Lett.*, vol. 9, no. 3, pp. 380–384, Mar. 2020.

[101] H.-C. Yang, S. Choi, and M.-S. Alouini, "Ultra-reliable low-latency transmission of small data over fading channels: A data-oriented analysis," *IEEE Commun. Lett.*, vol. 24, no. 3, pp. 515–519, Mar. 2020.

[102] Z. Zhang, J. Dai, M. Zeng, D. Liu, and S. Mao, "Scalable video caching for information centric wireless networks," *IEEE Access*, vol. 8, pp. 77272–77284, 2020.

[103] M. Zhang, B. Hao, R. Wang, and Y. Wang, "A pre-caching strategy based on the content relevance of smart device's request in information-centric IoT," *IEEE Access*, vol. 8, pp. 75761–75771, 2020.

[104] I. U. Din, H. Asmat, and M. Guizani, "A review of information centric network-based Internet of Things: Communication architectures, design issues, and research opportunities," *Multimedia Tools Appl.*, vol. 78, no. 21, pp. 30241–30256, Nov. 2019.

[105] L. Bracciale, P. Loreti, A. Detti, R. Paolillo, and N. B. Melazzi, "Lightweight named object: An ICN-based abstraction for IoT device programming and management," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5029–5039, Jun. 2019.

[106] X. Chen, C. Xu, M. Wang, T. Cao, L. Zhong, and G.-M. Muntean, "Optimal coded caching in 5G information-centric Device-to-Device communications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[107] B. Chen, L. Liu, and H. Ma, "HAC: Enable high efficient access control for information-centric Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10347–10360, Oct. 2020.

[108] S. Vural, P. Navaratnam, N. Wang, C. Wang, L. Dong, and R. Tafazolli, "In-network caching of Internet-of-Things data," in *Proc. IEEE ICC*, Jun. 2014, pp. 3185–3190.

[109] B. Chen, L. Liu, Z. Zhang, W. Yang, and H. Ma, "BRR-CVR: A collaborative caching strategy for information-centric wireless sensor networks," in *Proc. 12th Int. Conf. Mobile Ad-Hoc Sensor Netw. (MSN)*, Dec. 2016, pp. 31–38.

[110] D. D. Van and Q. Ai, "An efficient in-network caching decision algorithm for Internet of Things," *Int. J. Commun. Syst.*, vol. 31, no. 8, p. e3521, May 2018.

[111] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, and S. Alahmadi, "Cache freshness in named data networking for the Internet of Things," *Comput. J.*, vol. 61, no. 10, pp. 1496–1511, Jan. 2018.

[112] M. Naeem, R. Ali, B.-S. Kim, S. Nor, and S. Hassan, "A periodic caching strategy solution for the smart city in information-centric Internet of Things," *Sustainability*, vol. 10, no. 7, p. 2576, Jul. 2018.

[113] B. Chen, L. Liu, M. Sun, and H. Ma, "IoTCache: Toward data-driven network caching for Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10064–10076, Dec. 2019.

[114] Z. Zhang, C.-H. Lung, I. Lambadaris, and M. St-Hilaire, "IoT data lifetime-based cooperative caching scheme for ICN-IoT networks," in *Proc. IEEE ICC*, Kansas, MO, USA, May 2018, pp. 1–7.

[115] B. Wang, Y. Sun, S. Li, and Q. Cao, "Hierarchical matching with peer effect for low-latency and high-reliable caching in social IoT," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1193–1209, Feb. 2019.

[116] K. Hasan and S.-H. Jeong, "Efficient caching for data-driven IoT applications and fast content delivery with low latency in ICN," *Appl. Sci.*, vol. 9, no. 22, p. 4730, Nov. 2019.

[117] T.-A. Do, S.-W. Jeon, and W.-Y. Shin, "How to cache in mobile hybrid IoT networks?" *IEEE Access*, vol. 7, pp. 27814–27828, 2019.

[118] J. Pfender, A. Valera, and W. K. G. Seah, "Easy as ABC: A lightweight centrality-based caching strategy for information-centric IoT," in *Proc. 6th ACM Conf. Inf.-Centric Netw. (ICN)*, Sep. 2019, pp. 100–111.

[119] H. Wu, J. Li, J. Zhi, Y. Ren, and L. Li, "Edge-oriented collaborative caching in information-centric networking," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2019, pp. 1–6.

[120] S. Zhang and J. Liu, "Optimal probabilistic caching in heterogeneous IoT networks," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3404–3414, Apr. 2020.

[121] C. Long, Y. Cao, T. Jiang, and Q. Zhang, "Edge computing framework for cooperative video processing in multimedia IoT systems," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1126–1139, May 2018.

[122] A. Orsino, R. Kovalchukov, A. Samuylov, D. Moltchanov, S. Andreev, Y. Koucheryavy, and M. Valkama, "Caching-aided collaborative D2D operation for predictive data dissemination in industrial IoT," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 50–57, Jun. 2018.

[123] M.-O. Pahl, S. Liebald, and L. Wustrich, "Machine-learning based IoT data caching," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, Apr. 2019, pp. 9–12.

[124] Y. Hao, Y. Miao, L. Hu, M. S. Hossain, G. Muhammad, and S. U. Amin, "Smart-Edge-CoCaCo: AI-enabled smart edge with joint computation, caching, and communication in heterogeneous IoT," *IEEE Netw.*, vol. 33, no. 2, pp. 58–64, Mar. 2019.

[125] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge computing-supported Internet of Things," *IEEE Access*, vol. 7, pp. 69194–69201, Jun. 2019.

[126] L. Wang, H. Wu, Z. Han, P. Zhang, and H. V. Poor, "Multi-hop cooperative caching in social IoT using matching theory," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2127–2145, Apr. 2018.

[127] Y. Chen, X. Gong, R. Ou, L. Duan, and Q. Zhang, "Crowdcaching: Incentivizing D2D-enabled caching via coalitional game for IoT," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5599–5612, Jun. 2020.

[128] Akamai. *Over the Air (OTA)*. Accessed: Jul. 2020. [Online]. Available: https://www.akamai.com.

[129] Microsoft. *Microsoft Azure CDN*. Accessed: Jul. 2020. [Online]. Available: https://azure.microsoft.com/en-us/services/cdn/

[130] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the Internet of Things with edge computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan. 2018.

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

[132] A. Elgabli, H. Khan, M. Krouka, and M. Bennis, "Reinforcement learning based scheduling algorithm for optimizing age of information in ultra reliable low latency networks," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2019, pp. 1–6.

[133] A. Ksentini and P. A. Frangoudis, "Toward slicing-enabled multi-access edge computing in 5G," *IEEE Netw.*, vol. 34, no. 2, pp. 99–105, Mar. 2020.

[134] Y. Liu, Y. Zhou, J. Yuan, and L. Liu, "Delay aware flow scheduling for time sensitive fronthaul networks in centralized radio access network," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2992–3009, May 2020.

[135] Z. Ning, R. Y. K. Kwok, K. Zhang, X. Wang, M. S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, and B. Sadoun, "Joint computing and caching in 5G-envisioned Internet of vehicles: A deep reinforcement learning-based traffic control system," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 5, 2020, doi: 10.1109/TITS.2020.2970276.

[136] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive IoT networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4641–4654, May 2020.

[137] M. P. Véstias, R. P. Duarte, J. T. de Sousa, and H. C. Neto, "Moving deep learning to the edge," *Algorithms*, vol. 13, no. 5, p. 125, May 2020.

[138] M. Asad, A. Moustafa, and T. Ito, "FedOpt: Towards communication efficiency and privacy preservation in federated learning," *Appl. Sci.*, vol. 10, no. 8, p. 2864, Apr. 2020.

[139] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," 2016, *arXiv:1602.05629*. [Online]. Available: http://arxiv.org/abs/1602.05629
[140] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6532–6542, Oct. 2020.
[141] H.-K. Lim, J.-B. Kim, J.-S. Heo, and Y.-H. Han, "Federated reinforcement learning for training control policies on multiple IoT devices," *Sensors*, vol. 20, no. 5, p. 1359, Mar. 2020.
[142] H. Sun, S. Li, F. R. Yu, Q. Qi, J. Wang, and J. Liao, "Towards communication-efficient federated learning in the Internet of Things with edge computing," *IEEE Internet Things J.*, early access, May 15, 2020, doi: 10.1109/JIOT.2020.2994596.
[143] X. Wu, Z. Liang, and J. Wang, "FedMed: A federated learning framework for language modeling," *Sensors*, vol. 20, no. 14, p. 4048, Jul. 2020.
[144] A. Pustozerova and R. Mayer. *Information Leaks in Federated Learning*. Accessed: Oct. 2020. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2020/04/diss2020-23004-paper.pdf
[145] L. Wang, W. Wang, and B. Li, "CMFL: Mitigating communication overhead for federated learning," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 954–964.
[146] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," 2019, *arXiv:1905.06641*. [Online]. Available: http://arxiv.org/abs/1905.06641
[147] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5986–5994, Jul. 2020.
[148] Y. Sun, S. Zhou, and D. Gündüz, "Energy-aware analog aggregation for federated learning with redundant data," in *Proc. ICC*, Jun. 2020, pp. 7–11.
[149] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Convergence of federated learning over a noisy downlink," Aug. 2020, *arXiv:2008.11141*. [Online]. Available: http://arxiv.org/abs/2008.11141
[150] W. Zhang, T. Zhou, Q. Lu, X. Wang, C. Zhu, H. Sun, Z. Wang, S. Kit Lo, and F.-Y. Wang, "Dynamic fusion based federated learning for COVID-19 detection," Sep. 2020, *arXiv:2009.10401*. [Online]. Available: http://arxiv.org/abs/2009.10401
[151] J. Feng, C. Rong, F. Sun, D. Guo, and Y. Li, "PMF: A privacy-preserving human mobility prediction framework via federated learning," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 4, no. 1, pp. 1–21, Mar. 2020.
[152] P. Kairouz *et al.*, "Advances and open problems in federated learning," 2019, *arXiv:1912.04977*. [Online]. Available: http://arxiv.org/abs/1912.04977
[153] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140699–140725, Aug. 2020.
[154] Z. Du, C. Wu, T. Yoshinaga, K.-L.-A. Yau, Y. Ji, and J. Li, "Federated learning for vehicular Internet of Things: Recent advances and open issues," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 45–61, May 2020.
[155] T. Striffler, N. Michailow, and M. Bahr, "Time-sensitive networking in 5th generation cellular networks–current state and open topics," in *Proc. IEEE 2nd 5G World Forum (5GWF)*, Sep. 2019, pp. 547–552.
[156] D. Ginthor, J. von Hoyningen-Huene, R. Guillaume, and H. Schotten, "Analysis of multi-user scheduling in a TSN-enabled 5G system for industrial applications," in *Proc. IEEE Int. Conf. Ind. Internet (ICII)*, Nov. 2019, pp. 11–12.
[157] M. A. Ojewale, P. M. Yomsi, and B. Nikolic, "Multi-level preemption in TSN: Feasibility and requirements analysis," in *Proc. IEEE 23rd Int. Symp. Real-Time Distrib. Comput. (ISORC)*, May 2020, pp. 19–21.
[158] L. Velasco and M. Ruiz, "Supporting time-sensitive and best-effort traffic on a common metro infrastructure," *IEEE Commun. Lett.*, vol. 24, no. 8, pp. 1664–1668, Aug. 2020.
[159] A. Larranaga, M. C. Lucas-Estañ, I. Martinez, I. Val, and J. Gozalvez, "Analysis of 5G-TSN integration to support industry 4.0," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 8–11.

LILIAN CHARLES MUTALEMWA received the B.Eng. degree in telecommunications engineering from the University of Essex, Colchester, U.K., in 2008, and the M.Sc. degree in mobile and satellite communications from the University of Surrey, Guildford, U.K., in 2010. She is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Chosun University, Gwangju, South Korea. Since 2012, she has been with The Open University of Tanzania, Tanzania, where she is also an Assistant Lecturer with the Department of Information and Communication Technology. Her current research interests include wireless communication systems, 5G, the IoT, and network security and privacy.

SEOKJOO SHIN (Member, IEEE) received the B.Eng. degree in electronics engineering from Korea Aerospace University, South Korea, and the M.S. and Ph.D. degrees from the Department of Information and Communications, Gwangju Institute of Science and Technology (GIST), South Korea, in 1999 and 2002, respectively. He joined the Mobile Telecommunication Research Laboratory, Electronics and Telecommunications Research Institute (ETRI), South Korea, in 2002. In 2003, he joined the Faculty of Engineering, Chosun University, where he is currently a Full Professor with the Department of Computer Engineering. He spent 2009 as a Visiting Researcher at Georgia Tech, USA. His research interests include wireless communication systems, the IoT, AI related to networking, and network security and privacy.