Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence

Shuiguang Deng, *Senior Member, IEEE*, Hailiang Zhao, Weijia Fang, Jianwei Yin, Schahram Dustdar, *Fellow, IEEE*, and Albert Y. Zomaya, *Fellow, IEEE*

***Abstract*—Along with the rapid developments in communication technologies and the surge in the use of mobile devices, a brand-new computation paradigm, edge computing, is surging in popularity. Meanwhile, the artificial intelligence (AI) applications are thriving with the breakthroughs in deep learning and the many improvements in hardware architectures. Billions of data bytes, generated at the network edge, put massive demands on data processing and structural optimization. Thus, there exists a strong demand to integrate edge computing and AI, which gives birth to edge intelligence. In this article, we divide edge intelligence into AI for edge (intelligence-enabled edge computing) and AI on edge (artificial intelligence on edge). The former focuses on providing more optimal solutions to key problems in edge computing with the help of popular and effective AI technologies while the latter studies how to carry out the entire process of building AI models, i.e., model training and inference, on the edge. This article provides insights into this new interdisciplinary field from a broader perspective. It discusses the core concepts and the research roadmap, which should provide the necessary background for potential future research initiatives in edge intelligence.**

***Index Terms*—Computation offloading, edge computing, edge intelligence, Federated learning, wireless networking (WN).**

I. Introduction

COMMUNICATION technologies are undergoing a new revolution. 5G creates tremendous opportunities for social digitalization and industrial interconnection. In the 5G era, on top of the physical infrastructure, diversified service requirements (eMBB, mMTC, and uRLLC) can be met in the service-oriented end-to-end network slicing architecture [1]. The substantive characteristics of the network slicing architecture is *cloudification*, which involves the transformation from traditional hardbox network functions to *all-on-cloud* management plane. Here, the cloud not only refers to the regional datacenters but also the edge-cloud servers in the proximity of mobile service subscribers. With the proliferation of the Internet of Things (IoT), more data are created by widespread and geographically distributed mobile and IoT devices, and probably more than the data generated by the mega-scale cloud datacenters [2]. According to the prediction by Ericsson, 45% of the 40-ZB global Internet data will be generated by the IoT devices in 2024 [3]. Offloading such huge data from the edge to the cloud is intractable because it can lead to excessive network congestion. Therefore, a more applicable way is to handle user demands at the edge-cloud servers directly, which leads to the birth of a brand-new computation paradigm, edge computing [4]. In general, edge computing offers particular mobile services that need ultralow latency, high bandwidth, and real-time access to radio network information the deployment and management at the edge of network [5]. The subject of edge computing spans many concepts and technologies in diverse disciplines, including service-oriented computing (SOC), software-defined networking (SDN), and computer architecture, to name a few. The principle of edge computing is to push the computation and communication resources from the cloud to the edge of networks to provide services and perform computations, avoiding unnecessary communication latency and enabling faster responses for the end users. Edge computing is a booming field today.

No one can deny that artificial intelligence (AI) is developing rapidly nowadays. Big data processing necessitates that more powerful methods, i.e., AI technologies, for extracting insights that lead to better decisions and strategic business moves. In the last decade, with the huge success of AlexNet and deep neural networks (DNNs), which can learn the deep representation of data, have become the most popular machine learning architectures. Deep learning, represented by DNNs and their offshoots, i.e., convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs), has gradually become the most popular AI methods in the last few years. Deep learning has made striking breakthroughs in a wide spectrum of fields, including computer vision, speech recognition, natural language processing, and board games. Besides, hardware architectures and platforms keep on improving at a rapid rate,

Manuscript received September 2, 2019; revised November 10, 2019, February 6, 2020, and February 17, 2020; accepted March 28, 2020. Date of publication April 1, 2020; date of current version August 12, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1400601, in part by the National Science Foundation of China under Grant 61772461 and Grant 61825205, and in part by the Natural Science Foundation of Zhejiang Province under Grant LR18F020003. *(Corresponding author: Weijia Fang.)*

Shuiguang Deng is with the First Affiliated Hospital, Zhejiang University School of Medicine, Hangzhou 310003, China, and also with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310058, China (e-mail: dengsg@zju.edu.cn).

Hailiang Zhao and Jianwei Yin are with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310058, China (e-mail: hliangzhao@zju.edu.cn; zjuyjw@zju.edu.cn).

Weijia Fang is with the First Affiliated Hospital, Zhejiang University School of Medicine, Hangzhou 310003, China (e-mail: weijiafang@zju.edu.cn).

Schahram Dustdar is with the Distributed Systems Group, Technische Universität Wien, 1040 Vienna, Austria (e-mail: dustdar@dsg.tuwien.ac.at).

Albert Y. Zomaya is with the School of Computer Science, University of Sydney, Sydney, NSW 2006, Australia (e-mail: albert.zomaya@sydney.edu.au).

Digital Object Identifier 10.1109/JIOT.2020.2984887

TABLE I
RELATED SURVEYS AND THEIR EMPHASES

Perspectives	Related Surveys	Highlights
Intelligent Wireless Networking	[6] [7] [8] [9]	• Summarize the utilization of machine learning on the wireless edge • Including basic principles and general applications • Focus on resource management, networking, and mobility management • Optimization across different layers with machine learning technologies
Definitions and Divisions of Edge Intelligence	[10] [11] [12]	• Motivation, definition, division of Edge Intelligence • Including architectures, enabling technologies, learning frameworks, and software platforms • Focus on model training and inference on edge • Discuss the application scenarios and the practical implementations

which makes it possible to satisfy the requirements of the computation-intensive deep learning models. The application-specific accelerators are designed for further improvement in throughput and energy efficiency (EE). In conclusion, driven by the breakthroughs in deep learning and the upgrade of hardware architectures, AI is undergoing sustained success and development.

Considering that AI is functionally necessary for the quick analysis of huge volumes of data and extracting insights, there exists a strong demand to *integrate* edge computing and AI, which gives rise to edge intelligence. Edge intelligence is not the simple combination of edge computing and AI. The subject of edge intelligence is tremendous and enormously sophisticated, covering many concepts and technologies, which are interwoven together in a complex manner. Currently, there is no formal and internationally accepted definition of edge intelligence. To deal with the problem, some researchers put forward their definitions. For example, Zhou *et al.* [10] believed that the scope of edge intelligence should not be restricted to running the AI models solely on the edge servers or devices but in collaboration of edge and cloud. They define six levels of edge intelligence, from *cloud-edge co-inference* (level 1) to *all on-device* (level 6). Zhang *et al.* [11] defined edge intelligence as the capability to enable edges to execute AI algorithms.

Edge intelligence, currently in its early stage, is attracting more researchers and companies from all over the world. To disseminate the recent advances of edge intelligence, Zhou *et al.* [10] have conducted a comprehensive and concrete survey of the recent research efforts on edge intelligence. They survey the architectures, enabling technologies, systems, and frameworks from the perspective of AI models' training and inference. Some works also study the concept from the perspective of *AI-driven fog computing* [6], [7]. For example, Peng and Zhang comprehensively summarized the recent advances of the performance analysis and radio resource allocation in the fog-radio access networks (F-RANs). This survey presents the advanced edge cache and adaptive model selection schemes to improve spectral efficiency (SE) and EE [9]. The authors also survey the F-RANs from the perspectives of the system architecture and key techniques, where the latter includes transmission mode selection and interference suppression [13]. In addition, Mao *et al.* studied the state-of-the-art researches on the applications of deep learning algorithms for different network layers. In Table I, we summarize related survey papers on edge intelligence.

However, the material in edge intelligence spans an immense and diverse spectrum of literature, in origin and in nature, which is not fully covered by these surveys. Many concepts are still unclear and questions remain unsolved. The research process actually motivated us to write this article to shed some light and provide more insights with simple and clear classification. We propose to establish a broader vision and perspective. Specifically, we suggest to distinguish edge intelligence into *AI for edge* and *AI on edge*.

1) *AI for edge* is a research direction focusing on providing a better solution to constrained optimization problems in edge computing with the help of effective AI technologies. Here, AI is used to endow edge with more intelligence and optimality. Therefore, it can be understood as intelligence-enabled edge computing (IEC).
2) *AI on edge* studies how to run AI models on edge. It is a framework for running training and inference of AI models with device-edge-cloud synergy, which aims at extracting insights from massive and distributed edge data with the satisfaction of algorithm performance, cost, privacy, reliability, efficiency, etc. Therefore, it can be interpreted as AI on edge (AIE).

We commit ourselves elucidating edge intelligence to provide a broader vision and perspective. In Section II, we discuss the relation between edge computing and AI. In Section III, we demonstrate the research roadmap of edge intelligence concisely with a hierarchical structure. Sections IV and V elaborate the state-of-the-art and grand challenges on *AI for edge* and *AI on edge*, respectively. Section VI concludes this article.

II. RELATIONS BETWEEN EDGE COMPUTING AND AI

We believe that the confluence of AI and edge computing is natural and inevitable. In effect, there is an interactive relationship between them. On the one hand, AI provides edge computing with technologies and methods, and edge computing can unleash its potential and scalability with AI; on the other hand, edge computing provides AI with scenarios and platforms, and AI can expand its applicability with edge computing.

AI Provides Edge Computing With Technologies and Methods: In general, edge computing is a distributed computing paradigm, where software-defined networks are built to decentralize data and provide services with robustness and elasticity. Edge computing faces resource allocation problems in different layers, such as CPU cycle frequency, access jurisdiction, radio frequency, bandwidth, and so on. As a result, it has great demands on various powerful optimization tools to enhance system efficiency. AI technologies are capable to handle this task. Essentially, the AI models extract unconstrained optimization problems from real scenarios and then find the asymptotically optimal solutions iteratively with the SGD methods. Either statistical learning methods or deep learning methods can offer help and advice for the edge. Besides, reinforcement learning, including multiarmed bandit (MAB) theory, multiagent learning, and deep Q-network (DQN), is playing a growing and important role in resource allocation problems for the edge.

Edge Computing Provides AI With Scenarios and Platforms: The surge of IoT devices makes the Internet of Everything (IoE) a reality [14]. More data are created by widespread and geographically distributed mobile and IoT devices, other than the mega-scale cloud datacenters. Many more application scenarios, such as intelligent networked vehicles, autonomous driving, smart home, smart city, and real-time data processing in public security, can greatly facilitate the realization of AI from theory to practice. Besides, AI applications with high communication quality and low computational power requirements can be migrated from cloud to edge. In a word, edge computing provides AI with a heterogeneous platform full of rich capabilities. Nowadays, it is gradually becoming possible that AI chips with computational acceleration, such as field-programmable gate arrays (FPGAs), graphics processing units (GPUs), tensor processing units (TPUs), and neural processing units (NPUs), are integrated with intelligent mobile devices. More corporations participate in the design of chip architectures to support the edge computation paradigm and facilitate DNN acceleration on the resource-limited IoT devices. The hardware upgrade on edge also injects vigor and vitality into AI.

III. Research Roadmap of Edge Intelligence

The architectural layers in the edge intelligence roadmap, depicted in Fig. 1, describe a logical separation for the two directions, respectively, i.e., *AI for edge* (left) and *AI on edge* (right). In the bottom-up approach, we divide research efforts in edge computing into topology, content, and service. AI technologies can be utilized in all of them. By the top-down decomposition, we divide the research efforts in *AI on edge* into model adaptation, framework design, and processor acceleration. Before discussing *AI for edge* and *AI on edge* separately, we first describe the goal to be optimized for both of them, which is collectively known as Quality of Experience (QoE). QoE remains at the top of the roadmap.

A. Quality of Experience

We believe that QoE should be *application dependent* and determined by jointly considering multicriteria: performance, cost, privacy (security), efficiency, and reliability.

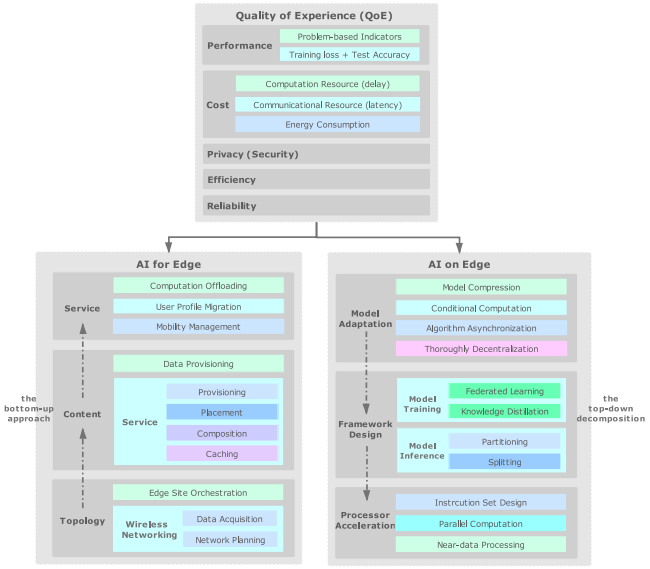

Fig. 1. Research roadmap of edge intelligence.

1) *Performance:* Ingredients of *performance* are different for *AI for edge* and *AI on edge*. As for the former, performance indicators are problem dependent. For example, performance could be *the ratio of successfully offloading* when it comes to the computation offloading problems. It could be the service providers' *need-to-be-maximized revenue* and *need-to-be-minimized hiring costs* of the base stations (BSs) when it comes into the service placement problems. As for the latter, performance mainly consists of training loss and inference accuracy, which are the most important criteria for the AI models. Although the computation scenarios have changed from the cloud clusters to the synergized system of device, edge, and cloud, these criteria still play important roles.
2) *Cost:* Cost usually consists of computation cost, communication cost, and energy consumption. Computation cost reflects the demand for computing resources, such as achieved CPU cycle frequency and allocated CPU time while communication cost presents the request for communication resources, such as power, frequency band, and access time. Many works also focused on minimizing the delay (latency) caused by allocated computation and communication resources. Energy consumption is not unique to edge computing but more crucial due to the limited battery capacity of mobile devices. Cost reduction is crucial because edge computing promises a dramatic reduction in delay and energy consumption by tackling the key challenges for realizing 5G.
3) *Privacy (Security):* With the increased awareness of the leaks of public data, privacy preservation has become one of the hottest topics in recent years. The status quo led to the birth of Federated learning, which aggregates

local machine learning models from distributed devices while preventing data leakage [15]. The security is closely tied with privacy preservation. It also has an association with the robustness of middleware and software of edge systems, which are not considered in this article.

4) *Efficiency:* Whatever *AI for edge* or *AI on edge*, high efficiency promises us a system with excellent performance and low overhead. The pursuit of efficiency is the key factor for improving the existing algorithms and models, especially for *AI on edge*. Many approaches, such as model compression, conditional computation, and algorithm asynchronization are proposed to improve the efficiency of training and inference of deep AI models.
5) *Reliability:* System reliability ensures that edge computing will not fail throughout any prescribed operating periods. It is an important indicator of user experience. For edge intelligence, system reliability appears to be particularly important for *AI on edge* because the model training and inference are usually carried out in a distributed and synchronized way and the participated local users have a significant probability of failing to complete the model upload and download due to wireless network congestion.

B. Recapitulation of IEC

The left side of the roadmap, depicted in Fig. 1, is *AI for edge*. We name this kind of work IEC (i.e., IEC) as AI provides powerful tools for solving complex learning, planning, and decision-making problems. By the bottom-up approach, the key concerns in edge computing are categorized into three layers, i.e., topology, content, and service.

For *topology*, we pay close attention to the orchestration of edge sites (OES) and wireless networking (WN). In this article, we define an edge site as a micro datacenter with applications deployed, attached to a small-cell BS (SBS). OES studies the deployment and installation of the wireless telecom equipment and servers. In recent years, research efforts on the management and automation of unmanned aerial vehicles (UAVs) became very popular [16]–[18]. UAVs with a small server and an access point can be regarded as moving edge servers with strong maneuverability. Therefore, many works explore scheduling and trajectory planning problems with the minimization of the energy consumption of UAVs. For example, Chen *et al.* studied the power consumption of UAVs by caching the popular contents under predictions, where a conceptor-based echo state network (ESN) algorithm is proposed to learn the mobility pattern of users. With the help of this effective machine learning technique, the proposed algorithm greatly outperforms benchmarks in terms of transmit power and QoE satisfaction. WN studies data acquisition and network planning. The former concentrates on the fast acquisition from rich but highly distributed data at subscribed edge devices while the latter concentrates on network scheduling, operation, and management. Fast data acquisition includes multiple access, radio resource allocation, and signal encoding/decoding. Network planning studies efficient management with protocols and middleware. In recent years, there has been an increasing trend in *intelligent* networking, which involves building an intelligent wireless communication mechanism by the popular AI technologies. For example, Zhu *et al.* [19] proposed *learning-driven communication*, which exploits the coupling between communication and learning in edge learning systems. In addition, Sun *et al.* [20] studied the resource management in F-RANs with deep reinforcement learning (DRL). In order to minimize long-term system power consumption, a Markov decision process (MDP) is formulated and the DQN technique is utilized to make intelligent decisions on the user's equipment communication modes [20].

For *content*, we place an emphasis on data provisioning, service provisioning, service placement, service composition, and service caching. For data and service provisioning, the available resources can be provided by remote cloud datacenters and edge servers. In recent years, there exist research efforts on constructing lightweight QoS-aware service-based frameworks [21]–[23]. The shared resources can also come from the mobile devices if a proper incentive mechanism is employed. Service placement is an important complement to service provisioning, which studies where and how to deploy complex services on possible edge sites. In recent years, many works studied service placement from the perspective of application service providers (ASPs). For example, Chen *et al.* [24] tried deploying services under a limited budget on basic communication and computation infrastructures. After that, the MAB theory, an embranchment of reinforcement learning, was adopted to optimize the service placement decision. Service composition studies how to select candidate services for composition in terms of energy consumption and QoE of the mobile end users [25]–[27]. It opens up research opportunities where the AI technologies can be utilized to generate better service selection schemes. Service caching can also be viewed as a complement to service provisioning. It studies how to design a caching pool to store the frequently visited data and services. Service caching can also be studied in a cooperative way [28]. It leads to research opportunities where multiagent learning can be utilized to optimize QoE in large-scale edge computing systems.

For *service*, we focus on computation offloading, user profile migration, and mobility management. Computation offloading studies the load balancing of various computational and communication resources in the manner of edge server selection and frequency spectrum allocation. More research efforts focus on dynamically managing the radio and computational resources for multiuser multiserver edge computing systems, utilizing the Lyapunov optimization techniques [29], [30]. In recent years, optimizing computation offloading decisions via DQN is popular [31]–[33]. It models the computation offloading problem as an MDP and maximizes the long-term utility performance. The utility can be composed of the above QoE indicators and evolves according to the iterative Bellman equation. After that, the asymptotically optimal computation offloading decisions are achieved based on DQN. The user profile migration studies how to adjust the place of user profiles (configuration files, private data, logs, etc.) when the mobile users are in constant motion.

The user profile migration is often associated with mobility management [34]. In [35], the proposed JCORM algorithm jointly optimizes computation offloading and migration by formulating cooperative networks. It opens research opportunities where more advanced AI technologies can be utilized to improve optimality. Many existing research efforts study mobility management from the perspective of statistics and probability theory. It has strong interests in realizing mobility management with AI.

C. Recapitulation of AIE

The right side of the roadmap is *AI on edge*. We name this kind of work AIE (i.e., AI on edge) since it studies how to carry out the training and inference of AI models on the network edge. By top-down decomposition, we divide the research efforts in *AI on edge* into three categories: 1) model adaptation; 2) framework design; and 3) processor acceleration. Considering that the research efforts in model adaptation are based on existing training and inference frameworks, let us introduce framework design in the first place.

1) Framework Design: Framework design aims at providing a better training and inference architecture for the edge without modifying the existing AI models. Researchers attempt to design new frameworks for both model training and model inference.

For Model Training: To the best of our knowledge, for model training, all proposed frameworks are distributed, except those knowledge distillation-based ones. The distributed training frameworks can be divided into data splitting and model splitting [36]. Data splitting can be further divided into master-device splitting, helper-device splitting, and device-device splitting. The differences lie where the training samples come from and how the global model is assembled and aggregated. Model splitting separates neural networks' layers and deploys them on different devices. It highly relies on sophisticated pipelines. The knowledge distillation-based frameworks may or may not be decentralized, and they rely on transfer learning technologies [37]. Knowledge distillation can enhance the accuracy of shallow student networks. It first trains a basic network on a basic data set. After that, the learned features can be transferred to student networks to be trained on their data sets, respectively. The basic network can be trained on the cloud or edge servers while those student networks can be trained by numerous mobile end devices with their private data, respectively. We believe that there exist great avenues to be explored in knowledge distillation-based frameworks for model training on the edge.

The most popular work in model training is *Federated learning* [15]. Federated learning is proposed to preserve privacy when training the DNNs in a distributed manner. Without aggregating the user private data to a central datacenter, Federated learning trains a series of local models on multiple clients. After that, a global model is optimized by averaging the trained gradients of each client. We are not going to elaborate on Federated learning thoroughly in this article. For more details please refer to [15]. For the edge nodes with limited storage and computing resource, it is unrealistic to train a comprehensive model on their own. Thus, a more applicable way is distributed training, where coordination between edge nodes is necessary. For the communication between edge nodes, the challenge is to optimize the global gradient from the distributed local models. No matter what learning algorithms are adopted, stochastic gradient descent (SGD) is necessary for model training. The distributed edge nodes use SGD to update their local gradients based on their own data set, which can be viewed as a minibatch. After that, they send their updated gradients to a central node for a global model upgrade. In this process, the tradeoffs between model performance and communication overhead has to be considered. If all edge nodes send their local gradients simultaneously, network congestion might be caused. A better approach is to selectively choose local gradients which have relatively large improvements. Under this circumstance, the model performance of the global model can be guaranteed while the communication overheads are reduced.

For Model Inference: Although model splitting is hard to realize for model training, it is a popular approach for model inference. Model splitting/partitioning can be viewed as a *framework* for model inference. Other approaches, such as model compression, input filtering, early exit, and so on, can be viewed as adaptations from the existing frameworks, which will be introduced in the next paragraph and elaborated on carefully in Section V-A. A typical example of model inference on edge is [38], where a DNN is split into two parts and carried out collaboratively. The computation-intensive part is running on the edge server while the other is running on the mobile device. The problem lies in where to split the layers and when to exit the intricate DNN according to the constraint on inference accuracy.

2) Model Adaptation: Model adaptation makes appropriate improvements based on the *existing* training and inference frameworks, usually Federated learning, to make them more applicable to the edge. Federated learning has the potential to run on the edge. However, the vanilla version of Federated learning has a strong demand for communication efficiency since full local models are supposed to be sent back to the central server. Therefore, many researchers exploit more efficient model updates and aggregation policies. Many works are devoted to reducing cost and increasing robustness while guaranteeing system performance. The methods to realize model adaptation include but not limited to model compression, conditional computation, algorithm asynchronization, and thorough decentralization. Model compression exploits the inherent sparsity structure of gradients and weights. Possible approaches include but not limited to quantization, dimensional reduction, pruning, precision downgrading, components sharing, cutoff, and so on. Those approaches can be realized by methods, such as the singular value decomposition (SVD), the Huffman coding, the principal component analysis (PCA), and several others. Conditional computation is an alternative way to reduce the amount of calculation by selectively turning off some unimportant calculations of DNNs. Possible approaches include but not limited to components shutoff, input filtering, early exit, results caching, and so on. Conditional computation can be viewed as blockwise

dropout [39]. Besides, random gossip communication can be utilized to reduce unnecessary calculations and model updates. The algorithm asynchronization tries aggregating local models in an asynchronous way. It is designed for overcoming the inefficient and lengthy synchronous steps of model updates in Federated learning. Thoroughly decentralization removes the central aggregator to avoid any possible leakage and address the central server's malfunction. The ways to achieve totally decentralization include but not limited to blockchain technologies and game-theoretical approaches.

3) Processor Acceleration: Processor acceleration focuses on structure optimization of DNNs in that the frequently used computation-intensive multiply-and-accumulate operations can be improved. The approaches to accelerate DNN computation on hardware include: 1) designing special instruction sets for DNN training and inference; 2) designing highly paralleled computing paradigms; and 3) moving computation closer to memory (near-data processing). Highly parallelized computing paradigms can be divided into temporal and spatial architectures [40]. The former architectures, such as CPUs and GPUs can be accelerated by reducing the number of multiplications and increasing throughput. The latter architectures can be accelerated by increasing data reuse with data flows. For example, Lee *et al.* [41] proposed an algorithm to accelerate CNN inference. The proposed algorithm converts a set of pretrained weights into values under given precision. It also puts near-data processing into practice with an adaptive implementation of the memristor crossbar arrays. In the research area of edge computing, a lot of works hammer at the co-design of model adaptation and processor acceleration. Considering that processor acceleration is mainly investigated by AI researchers, this article will not delve into it. More details on hardware acceleration for DNN processing can be found in [40].

IV. AI for Edge

In Section III-B, we divide the key issues in edge computing into three categories: 1) topology; 2) content; and 3) service. It just presents a classification and possible research directions but does not provide an in-depth analysis of how to apply AI technologies to the edge to generate more optimal solutions. This section will remedy this, Fig. 2 gives an example of how the AI technologies are utilized in the mobile-edge computing (MEC) environment. First, we need to identify the problem to be studied. Take performance optimization as an example, the optimization goal, decision variables, and potential constraints need to be confirmed. The need-to-be optimized goal could be the combination of task execution delay, transmission delay, and task dropping cost. The studied task can be either binary or partial. After that, the mathematical model should be constructed. If the long-term stability of the system is considered, the Lyapunov optimization technique cloud be used to formalize the problem. Finally, we should design an algorithm to solve the problem. In fact, the model construction is not only decided by the to-be-studied problem but also the to-be-applied optimization algorithms. Take DQN for example, we have to model the problem as an MDP with finite *states* and *actions*. Thus, the constraints cannot exist in the long-term optimization problem. The most common way is transferring those constraints into the penalty and adding the penalty to the optimization goal.

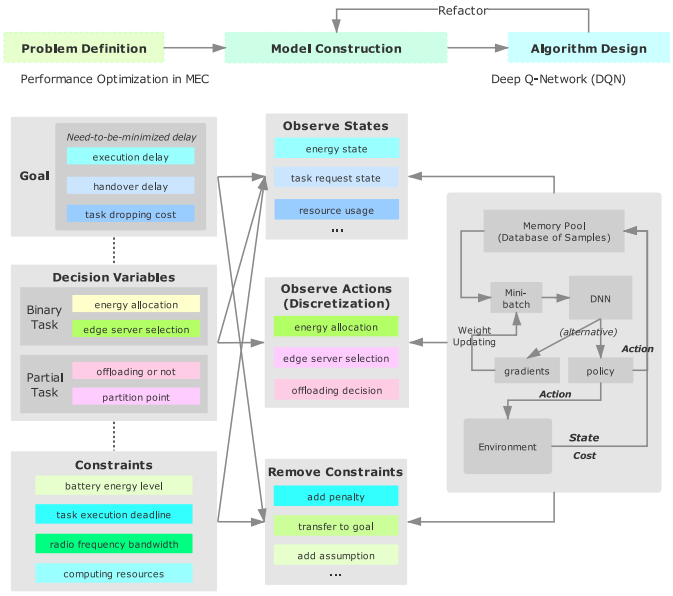

Fig. 2. Utilization of the AI technology for performance optimization.

Considering that current research efforts on *AI for edge* concentrate on WN, service placement, service caching, and computation offloading, we only focus on these topics in the following section. For research directions that have not been explored yet, we are expecting to see more works in due course.

A. State of the Art

1) Wireless Networking: The 5G technology promises eMBB, URLLC, and mMTC in a real-time and highly dynamic environment. Under the circumstances, researchers reach a consensus on that AI technologies should and can be integrated across the wireless infrastructure and mobile users [8]. We believe that AI should be synergistically applied to achieve intelligent network optimization in a fully online manner. One of the typical works in this area is [19]. This article advocates a new set of design principles for wireless communication on edge with machine learning technologies and models embedded, which are collectively named as *learning-driven communication*. It can be achieved across the whole process of data acquisition, which is in turn multiple access, radio resource management, and signal encoding.

Learning-driven multiple access advocates that the unique characteristics of wireless channels should be exploited for functional computation. Over-the-air computation (AirComp) is a typical technique used to realize it [42], [43]. Zhu *et al.* [44] put this principle into practice based on broadband analog aggregation (BAA). Concretely, Zhu *et al.* [44] suggested that the simultaneously transmitted model updates in Federated learning should be analog aggregated by exploiting the waveform-superposition property of multiaccess channels.

The proposed BAA can dramatically reduce communication latency compared with the traditional orthogonal frequency-division multiple access (OFDMA). The work in [45] explores the *over-the-air computation* for model aggregation in Federated learning. More specifically, Yang *et al.* [45] put the principle into practice by modeling the device selection and beamforming design as a sparse and low-rank optimization problem, which is computationally intractable. To solve the problem with a fast convergence rate, this article proposed a difference-of-convex-functions (DC) representation via successive convex relaxation. The numerical results show that the proposed algorithm can achieve lower training loss and higher inference accuracy compared with the state-of-the-art approaches. This contribution can also be categorized as model adaptation in *AI on edge*, but it accelerates Federated learning from the perspective of fast data acquisition.

Learning-driven radio resource management promotes the idea that radio resources should be allocated based on the value of transmitted data, not just the efficiency of spectrum utilization. Therefore, it can be understood as *importance-aware resource allocation* and an obvious approach is *importance-aware retransmission*. Liu *et al.* [46] put the principle into practice. This article proposed a retransmission protocol, named importance-aware automatic-repeat-request (importance ARQ). Importance ARQ makes the tradeoff between signal-to-noise ratio (SNR) and data uncertainty under the desired learning accuracy. It can achieve fast convergence while avoiding learning performance degradation caused by the channel noise.

Learning-driven signal encoding stipulates that signal encoding should be designed by jointly optimizing feature extraction, source coding, and channel encoding. A work that puts this principle into practice is [47], which proposes a hybrid-federated distillation (HFD) scheme based on separate source-channel coding and *over-the-air* computing. It adopts sparse binary compression with error accumulation in source-channel coding. For both digital and analog implementations over the Gaussian multiple-access channels, HFD can outperform the vanilla version of Federated learning in a poor communication environment. This principle has something in common with dimensional reduction and quantization from model adaptation in *AI on edge*, but it reduces the feature size from the source of data transmission. It opens up great research opportunities for the co-design of learning frameworks and data encoding.

Apart from *learning-driven communication*, some works contribute to *AI for WN* from the perspective of energy consumption and radio resource efficiency. Zhang *et al.* [48] proposed a DRL-based decentralized algorithm to maximize the sum capacity of vehicle-to-infrastructure users while meeting the latency and reliability requirements of vehicle-to-vehicle (V2V) pairs. Lu *et al.* [49] designed a deep RL-based energy trading algorithm to address the supply–demand mismatch problem for a smart grid with a large number of microgrids (MGs) without relying on the renewable energy generation and power demand models of other MGs. Shen *et al.* [50] utilized the graph neural networks (GNNs) to develop scalable methods for power control in K-user interference channels. This article first models the K-user interference channel as a complete graph, then it learns the optimal power control with a graph CNN. Temesgene *et al.* [51] studied an energy minimization problem where the baseband processes of the virtual small cells powered solely by energy harvesters and batteries can be opportunistically executed in a grid-connected edge server. Based on multiagent learning, several distributed fuzzy Q-learning-based algorithms are tailored. This article can be viewed as an attempt for coordinating with broadcasting. As we will expound later, some works on WN are often combined with computation offloading when they are studied in the form of optimization. State-of-the-art of these works is listed in Section IV-A3.

2) Service Placement and Caching: Many researchers study service placement from the perspective of ASPs. They model the data and service (it can be compounded and complex) placement problem as an MDP and utilize the AI methods such as reinforcement learning to achieve an optimal placement decision. A typical work implementing this idea is [24]. This article proposes a spatial–temporal algorithm based on MAB and achieves the optimal placement decisions while learning the benefit. Concretely, it studies how many SBSs should be rented for edge service hosting to maximize the expected utility up to a finite-time horizon. The expected utility is composed of delay reduction of all mobile users. After that, an MAB-based algorithm, named SEEN, is proposed to learn the local users' service demand patterns of SBSs. It can achieve the balance between *exploitation* and *exploration* automatically according to the fact that whether the set of SBSs is chosen before. Another work that attempts to integrate AI approaches with service placement is [52]. This article jointly decides which SBS to deploy each data block and service component and how much harvested energy should be stored in mobile devices with a DQN-based algorithm. This article will not elaborate on DQN. More details can be found in [53].

Service caching can be viewed as a complement to service placement [54]. The edge servers can be equipped with special *service cache* to satisfy the user demands on popular content. A wide range of optimization problems on service caching are proposed to endow edge servers with learning capability. Zhang *et al.* [28] studied a sequential fetch-cache decision based on dynamic prices and user requests. This article endows SBSs with efficient fetch-cache decision-making schemes operating in dynamic settings. Concretely, it formulates a cost minimization problem with service popularity considered. For the long-term stochastic optimization problem, several computationally efficient algorithms are developed based on Q-learning.

3) Computation Offloading: Computation offloading can be considered as the most active topic when it comes to *AI for edge*. It studies the transfer of resource-intensive computational tasks from resource-limited mobile devices to edge or cloud. This process involves the allocation of many resources, ranging from CPU cycles to channel bandwidth. Therefore, the AI technologies with strong optimization abilities have been extensively used in recent years. Among all these AI

technologies, Q-learning and its derivatives, DQN, are in the spotlight. For example, Qiu *et al.* [55] designed a Q-learning-based algorithm for computation offloading. It formulates the computation offloading problem as a noncooperative game in multiuser multiserver edge computing systems and proves that the Nash equilibrium exists. Then, this article proposes a model-free Q-learning-based offloading mechanism which helps the mobile devices learn their long-term offloading strategies to maximize their long-term utilities.

More works are based on DQN because *the curse of dimensionality* could be overcome with nonlinear function approximation. For example, Min *et al.* [31] studied the computation offloading for IoT devices with energy harvesting in multiserver MEC systems. The *need-to-be-maximized* utility formed from overall data sharing gains, task dropping penalty, energy consumption, and computation delay, which is updated according to the Bellman equation. After that, DQN is used to generate the optimal offloading scheme. In [32] and [56], the computation offloading problem is formulated as an MDP with *finite* states and actions. The state set is composed of the channel qualities, the energy queue, and the task queue while the action set is composed of offloading decisions in different time slots. Then, a DQN-based algorithm is proposed to minimize the long-term cost. Based on DQN, task offloading decisions and wireless resource allocation are jointly optimized to maximize the data acquisition and analysis capability of the network [57], [58]. The work in [59] studies the knowledge-driven service offloading problem for vehicle of Internet. The problem is also formulated as a long-term planning optimization problem and solved based on DQN. In summary, computation offloading problems in various industrial scenarios have been extensively studied from all sorts of perspectives.

There also exist works which explore the task offloading problem with other AI technologies. For example, Yang *et al.* [60] proposed a long short-term memory (LSTM) network to predict the task popularity and then formulated a joint optimization of the task offloading decisions, computation resource allocation, and caching decisions. After that, a Bayesian learning automata-based multiagent learning algorithm is proposed for optimality.

B. Grand Challenges

Although it is popular to apply AI methods to edge for the generation of better solutions, however, there have been many challenges. In the next several sections, we list grand challenges across the whole theme of *AI for edge* research. These challenges are closely related but each has its own emphasis.

1) Model Establishment: If we want to use AI methods, the mathematical models have to be limited and the formulated optimization problem needs to be restricted. On the one hand, this is because the optimization basis of AI technologies, SGD, and minibatch gradient descent (MBGD) methods may not work well if the original search space is constrained. On the other hand, especially for MDPs, the state set and action set cannot be infinite, and discretization is necessary to avoid the curse of dimensionality before further processing. The common solution is changing the constraints into a penalty and incorporating them into the global optimization goal. The status quo greatly restricts the establishment of mathematical models which leads to performance degradation. It can be viewed as a compromise for the utilization of AI methods. Therefore, how to establish an appropriate system model poses great challenges.

2) Algorithm Deployment: The state-of-the-art works often formulate a combinatorial and NP-hard optimization problem which has fairly high computational complexity. Very few works can achieve an analytic approximate optimal solution with convex optimization methods. Actually, for *AI for edge*, the solution mostly comes from iterative learning-based approaches. There are many challenges that face when these methods are deployed on the edge in an online manner. Besides, another ignored challenge is which edge device should undertake the responsibility for deploying and running the proposed complicated algorithms. The existing research efforts usually concentrate on their specific problems and do not provide the details on that.

3) Balance Between Optimality and Efficiency: Although AI technologies can indeed provide solutions that are optimal, the tradeoff between optimality and efficiency cannot be ignored when it comes to the resource-constrained edge. Thus, how to improve the usability and efficiency of edge computing systems for different application scenarios with AI technologies embedded is a severe challenge. The tradeoff between optimality and efficiency should be realized based on the characteristics of dynamically changing requirements on QoE and the network resource structure. Therefore, it is coupling with the service subscribers' pursuing superiority and the utilization of available resources.

V. AI ON EDGE

In Section III-C, we divide the research efforts for *AI on edge* into model adaptation, framework design, and processor acceleration. The existing frameworks for model training and inference are rare. The training frameworks include Federated learning and knowledge distillation while the inference frameworks include model spitting and model partitioning. AI models on edge are by far limited when compared to cloud-based predictions because of the relatively limited compute and storage abilities. How to carry out the model training and inference on resource-scarce devices is a serious issue. As a result, compared with designing new frameworks, researchers in edge computing are more interested in improving the existing frameworks to make them more appropriate for the edge, usually reducing resource occupation. As a result, model adaptation based on Federated learning is prosperously developed. As we have mentioned earlier, processor acceleration will not be elaborated in detail. Therefore, we only focus on model adaptation in the following section. Table II lists the methods and the correlated papers. Their contributions are also highlighted.

TABLE II
METHODS AND THE CORRESPONDING PAPERS

Methods	Related Papers
Model Compression	• Sketched updates & structured updates [61] • Communication-efficient secure aggregation [62] • Mixed low-bitwidth compression [63] • Retraining-after-pruning [64] • Compressed RNN (based on Hybrid Matrix Decomposition) [65] • Binary Neural Networks (BNNs) [66] [67] [68] • ProNN (based on Stochastic Neighborhood Compression) [69]
Conditional Computation	• Runtime-throttleable block-level gating [70]
Algorithm Asynchronization	• GoSGD (based on Random-gossip communication) [71] • GossipGraD (based on Random-gossip communication) [72]
Thoroughly Decentralization	• BlockFL (based on Blockchain) [73] • Game-theoretical approach

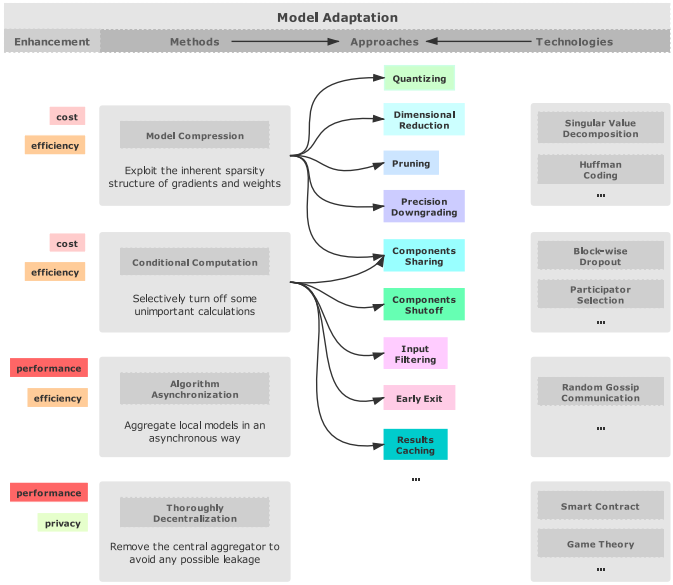

Fig. 3. Methods, approaches, and technologies of model adaptation.

A. State of the Art

1) Model Compression: As demonstrated in Fig. 3, the approaches for model compression include quantization, dimensionality reduction, pruning, components sharing, precision downgrading, and so on. They exploit the inherent sparsity structure of gradients and weights to reduce the memory and channel occupation as much as possible. The technologies to compress and quantize weights include but not limited to SVD, Huffman coding, and PCA. This article will not provide a thorough introduction to these due to limited space. Considering that many works simultaneously utilize the approaches mentioned above, we do not further divide the state-of-the-art in model compression. One more thing should be clearly noted is that model compression is suitable for both model training and model inference. Thus, we do not deliberately distinguish between them.

As we have mentioned earlier, communication efficiency is of the utmost importance for Federated learning. Minimizing the number of rounds of communication is the principal goal when we move Federated learning to the edge because updating the global model might not be achieved if one or more local devices are offline or the network is congested. Therefore, a lot of works focus on reducing the communication overhead for Federated learning from various perspectives. Compressing the trained models without reducing the inference accuracy is one of the best ways to realize it. For example, in [61], *structured updates* and *sketched updates* are proposed for reducing the uplink communication costs. For *structured updates*, the local update is learned from a restricted lower dimensional space; for *sketched updates*, the uploading model is compressed before sending to the central server. Bonawitz *et al.* [62] designed a communication-efficient secure aggregation protocol for high-dimensional data. The protocol can tolerate up to 33.3% of participating devices failing to complete the protocol, i.e., the system is robust. The work in [64] suggests that DNNs are typically overparameterized and their weights have significant redundancy. Meanwhile, pruning compensates for the loss in performance. Thus, this article proposes a *retraining-after-pruning* scheme. It retrains the DNN on new data while the pruned weights stay constant. The scheme can reduce the resource occupation while guaranteeing learning accuracy. The work in [63] exploits mixed low-bitwidth compression. It works on determining the minimum bit precision of each activation and weight under the given constraints on memory. Venkatesan and Li [66] used the binarized neural networks (BNNs), which have binary weights and activations to replace regular DNNs. This is a typical exploration of quantization. Analogously, Chakraborty *et al.* [67] proposed the hybrid network architectures combining binary and full-precision sections to achieve significant EE and memory compression with performance guaranteed. Thakker *et al.* [65] studied a compressed RNN cell implementation called the hybrid matrix decomposition (HMD) for model inference. It divides the matrix of network weights into two parts: 1) an unconstrained upper half and 2) a lower half composed of rank-1 blocks. The output features are composed of the rich part (upper) and the barren part (lower). This is an imaginative variation on compression, compared with traditional pruning or quantization. The numerical results show that it can not only achieve a faster runtime than pruning and but also retain more model accuracy than matrix factorization.

Some works also explore model compression based on partitioned DNNs. For example, Li *et al.* [74] proposed an auto-tuning neural network quantization framework for collaborative inference between edge and cloud. First, DNN is partitioned. The first part is quantized and executed on the edge devices while the second part is executed in cloud *with full precision*. The work in [75] proposes a framework to accelerate and compress model training and inference. It partitions DNNs into multiple sections according to their depth and constructs classifiers upon the intermediate features of different sections. Besides, the accuracy of classifiers is enhanced by knowledge distillation.

Apart from Federated learning, there exist works that probe into the execution of statistical learning models or other popular deep models, such as ResNet and VGG architectures on resource-limited end devices. For example, Gupta *et al.* [69] proposed ProtoNN, a compressed and accurate k-nearest neighbor (kNN) algorithm. ProtoNN learns a small number of prototypes to represent the entire training set by stochastic neighborhood compression (SNC) [76], and then projects the entire data in a lower dimension with a sparse projection matrix. It jointly optimizes the projection and prototypes with the explicit model size constraint. Chakraborty *et al.* [68] proposed the Hybrid-Net which has both binary and high-precision layers to reduce the degradation of learning performance. Innovatively, this article leverages PCA to identify significant layers in a binary network, other than dimensionality reduction. The *significance* here is identified based on the ability of a layer to expand into higher dimensional space.

Model compression is currently a very active direction in *AI on edge* because it is easy to implement. However, the state-of-the-art works are usually not tied to specific application scenarios of edge computing systems. There are opportunities for new works that construct edge platforms and hardware.

2) Conditional Computation: As demonstrated in Fig. 3, the approaches for conditional computation include components sharing, components shutoff, input filtering, early exit, results caching, and so on. To put it simply, conditional computation is selectively turning off some unimportant calculations. Thus, it can be viewed as blockwise dropout [39]. A lot of works devote themselves to ranking and selecting the most worthy part for computation or early stop if the confident threshold is achieved. For example, Hostetler [70] instantiated a *runtime-throttleable* neural network which can adaptively balance learning accuracy and resource occupation in response to a control signal. It puts conditional computation into practice via block-level gating.

This idea can also be put into use for participator selection. It selects the most valuable participators in Federated learning for model updates. The valueless participators will not engage the aggregation of the global model. To the best of our knowledge, currently, there is no work dedicated to participator selection. We are eagerly looking forward to exciting works on it.

3) Algorithm Asynchronization: As demonstrated in Fig. 3, algorithm asynchronization attempts to aggregate local models in an asynchronous way for Federated learning. As we have mentioned before, the participating local users have a significant probability of failing to complete the model upload and download due to the wireless network congestion. Apart from model compression, another way is exchanging weights and gradients *peer-to-peer* to reduce the high concurrency on wireless channels. Random-gossip communication is a typical example. Based on randomized gossip algorithms, Blot *et al.* [71] proposed GoSGD to train DNNs asynchronously. The most challenging problem for gossip training is the degradation of convergence rate in large-scale edge systems. To overcome the issue, Daily *et al.* [72] introduced GossipGraD, which can reduce the communication complexity greatly to ensure the fast convergence.

4) Thorough Decentralization: As demonstrated in Fig. 3, thorough decentralization attempts to remove the central aggregator to avoid any possible leakage. Although Federated learning does not require consumers' private data, the model updates still contain private information as some trust of the server coordinating the training is still required. To avoid privacy leaks altogether, the blockchain technology and game-theoretical approaches can assist in total decentralization.

By leveraging blockchain, especially smart contracts, the central server for model aggregating is not needed anymore. As a result, collapse triggered by model aggregation can be avoided. Besides, user privacy can be protected. We believe that the blockchain-based Federated learning will become a hot field and prosperous direction in the coming years. There exist works that put it into practice. In [73], the proposed blockchain-based federated learning architecture, BlockFL, takes edge nodes as miners. The miners exchange and verify all the local model updates contributed by each device and then run the Proof of Work (PoW). The miner who first completes the PoW generates a new block and receives the mining reward from the blockchain network. Finally, each device updates its local model from the freshest block. In this article, blockchain is effectively integrated with Federated learning to build a trustworthy edge learning environment.

B. Grand Challenges

The grand challenges for AI on edge are listed from the perspective of data availability, model selection, and coordination mechanism, respectively.

1) Data Availability: The toughest challenge lies in the availability and usability of raw training data because usable data are the beginning of everything. First, a proper incentive mechanism may be necessary for data provisioning from the mobile users. Otherwise, the raw data may not be available for model training and inference. Besides, the raw data from various end devices could have an obvious bias, which can greatly affect the learning performance. Although Federated learning can overcome the problem caused by non-i.i.d. samples to a certain extent, the training procedure still faces great difficulties in the design of robust communication protocol. Therefore, there are huge challenges in terms of data availability.

2) Model Selection: Presently, the selection of *need-to-be-trained* AI models faces severe challenges in the following aspects, across from the models themselves to the training

frameworks and hardware. First, how to select the befitting threshold of learning accuracy and scale of AI models for quick deployment and delivery. Second, how to select probe training frameworks and accelerator architectures under the limited resources. Model selection is coupling with resource allocation and management, thus the problem is complicated and challenging.

3) Coordination Mechanism: The proposed methods on model adaptation may not be pervasively serviceable because there could be a huge difference in computing power and communication resources between the heterogeneous edge devices. It may lead to that the same method achieves different learning results for different clusters of mobile devices. Therefore, the compatibility and coordination between the heterogeneous edge devices are of great essence. A flexible coordination mechanism between cloud, edge, and device in both hardware and middleware is imperative and urgently needed to be designed. It opens up research opportunities on a uniform API interface on edge learning for ubiquitous edge devices.

VI. Conclusion

Edge intelligence, although still in its early stages, has attracted more and more researchers and companies to get involved in studying and using it. This article attempts to provide possible research opportunities through a succinct and effective classification. Concretely, we first discuss the relation between edge computing and AI. We believe that they promote and reinforce each other. After that, we divide edge intelligence into *AI for edge* and *AI on edge* and sketch the research roadmap. The former focuses on providing a better solution to the key concerns in edge computing with the help of popular and rich AI technologies while the latter studies how to carry out the training and inference of AI models, on edge. Either *AI for edge* or *AI on edge*, the research roadmap is presented in a hierarchical architecture. By the bottom-up approach, we divide research efforts in edge computing into topology, content, and service and introduce some examples on how to energize edge with intelligence. By top-down decomposition, we divide the research efforts in *AI on edge* into model adaptation, framework design, and processor acceleration and introduce some existing research results. Finally, we present the state-of-the-art and grand challenges in several hot topics for both *AI for edge* and *AI on edge*. We attempted to provide some enlightening thoughts on the emerging field of edge intelligence. We hope that this article can stimulate fruitful discussions on potential future research directions for edge intelligence.

References

[1] GPAW Group. (Jun. 2019). *View on 5G Architecture: Version 3.0*. [Online]. Available: https://5g-ppp.eu/5g-ppp-architecture-public-consultation/

[2] M. Asif-Ur-Rahman *et al.*, "Toward a heterogeneous mist, fog, and cloud-based framework for the Internet of healthcare things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4049–4062, Jun. 2019.

[3] Ericsson. (2019). *IoT Connections Outlook: NB-IoT and CAT-M Technologies Will Account for Close to 45 Percent of Cellular IoT Connections in 2024*. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/june-2019/iot-connections-outlook

[4] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[5] EGM. (Nov. 2019). *Multi-Access Edge Computing (MEC): Study on MEC Support for Alternative Virtualization Technologies, V2.1.1*. [Online]. Available: https://www.etsi.org/deliver/etsi_gr/MEC/001_099/027/02.01.01_60/gr_MEC027v020101p.pdf

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

[7] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.

[8] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 4th Quart., 2019.

[9] M. Peng and K. Zhang, "Recent advances in fog radio access networks: Performance analysis and radio resource allocation," *IEEE Access*, vol. 4, pp. 5003–5009, 2016.

[10] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[11] X. Zhang, Y. Wang, S. Lu, L. Liu, L. Xu, and W. Shi, "OpenEI: An open framework for edge intelligence," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Dallas, TX, USA, 2019, pp. 1840–1851.

[12] Y. Han, X. Wang, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, early access, Jan. 30, 2020, doi: 10.1109/COMST.2020.2970550.

[13] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 46–53, Jul. 2016.

[14] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[15] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2016, pp. 1273–1282.

[16] J. Xu, Y. Zeng, and R. Zhang, "UAV-enabled wireless power transfer: Trajectory design and energy optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5092–5106, Aug. 2018.

[17] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.

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

[19] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Towards an intelligent edge: Wireless communication meets machine learning," 2018. [Online]. Available: arXiv:1809.00343.

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

[21] H. Wu *et al.*, "Revenue-driven service provisioning for resource sharing in mobile cloud computing," in *Service-Oriented Computing*, M. Maximilien, A. Vallecillo, J. Wang, and M. Oriol, Eds. Cham, Switzerland: Springer Int., 2017, pp. 625–640.

[22] S. Deng, Z. Xiang, J. Yin, J. Taheri, and A. Y. Zomaya, "Composition-driven IoT service provisioning in distributed edges," *IEEE Access*, vol. 6, pp. 54258–54269, 2018.

[23] S. Deng *et al.*, "Optimal application deployment in resource constrained distributed edges," *IEEE Trans. Mobile Comput.*, early access, Jan. 30, 2020, doi: 10.1109/TMC.2020.2970698.

[24] L. Chen, J. Xu, S. Ren, and P. Zhou, "Spatio-temporal edge service placement: A bandit learning approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 8388–8401, Dec. 2018.

[25] S. Deng, H. Wu, W. Tan, Z. Xiang, and Z. Wu, "Mobile service selection for composition: An energy consumption perspective," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 3, pp. 1478–1490, Jul. 2017. [Online]. Available: https://doi.org/10.1109/TASE.2015.2438020

[26] S. Deng, L. Huang, J. Taheri, J. Yin, M. Zhou, and A. Y. Zomaya, "Mobility-aware service composition in mobile communities," *IEEE*

Trans. Syst., Man, Cybern., Syst., vol. 47, no. 3, pp. 555–568, Mar. 2017. [Online]. Available: https://doi.org/10.1109/TSMC.2016.2521736

[27] Z. Wu, J. Yin, S. Deng, J. Wu, Y. Li, and L. Chen, "Modern service industry and crossover services: Development and trends in China," *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 664–671, Sep./Oct. 2016. [Online]. Available: https://doi.org/10.1109/TSC.2015.2418765

[28] 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.

[29] H. Zhao, W. Du, W. Liu, T. Lei, and Q. Lei, "QoE aware and cell capacity enhanced computation offloading for multi-server mobile edge computing systems with energy harvesting devices," in *Proc. IEEE Int. Conf. Ubiquitous Intell. Comput.*, Oct. 2018, pp. 671–678.

[30] H. Zhao, S. Deng, C. Zhang, W. Du, Q. He, and J. Yin, "A mobility-aware cross-edge computation offloading framework for partitionable applications," in *Proc. IEEE Int. Conf. Web Services*, Jul. 2019, pp. 193–200.

[31] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930–1941, Feb. 2019.

[32] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Performance optimization in mobile-edge computing via deep reinforcement learning," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–6.

[33] S. Deng *et al.*, "Dynamical resource allocation in edge for trustable IoT systems: A reinforcement learning method," *IEEE Trans. Ind. Informat.*, early access, Feb. 18, 2020, doi: 10.1109/TII.2020.2974875.

[34] S. Deng, H. Wu, D. Hu, and J. L. Zhao, "Service selection for composition with QoS correlations," *IEEE Trans. Services Comput.*, vol. 9, no. 2, pp. 291–303, Mar./Apr. 2016. [Online]. Available: https://doi.org/10.1109/TSC.2014.2361138

[35] C. Zhang, H. Zhao, and S. Deng, "A density-based offloading strategy for IoT devices in edge computing systems," *IEEE Access*, vol. 6, pp. 73520–73530, 2018.

[36] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," 2018. [Online]. Available: arxiv:1812.02858.

[37] J. Wang, J. Zhang, W. Bao, X. Zhu, B. Cao, and P. S. Yu, "Not just privacy: Improving performance of private deep learning in mobile cloud," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining*, 2018, pp. 2407–2416, doi: 10.1145/3219819.3220106.

[38] E. Li, Z. Zhou, and X. Chen, "Edge intelligence: On-demand deep learning model co-inference with device-edge synergy," in *Proc. Workshop Mobile Edge Commun. (MECOMM@SIGCOMM)*, Budapest, Hungary, Aug. 2018, pp. 31–36.

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.

[40] V. Sze, Y. Chen, T. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.

[41] J. Lee, J. K. Eshraghian, K. Cho, and K. Eshraghian, "Adaptive precision CNN accelerator using radix-X parallel connected memristor crossbars," 2019. [Online]. Available: arXiv.abs/1906.09395.

[42] O. Abari, H. Rahul, and D. Katabi, "Over-the-air function computation in sensor networks," 2016. [Online]. Available: arxiv:1612.02307.

[43] G. Zhu, L. Chen, and K. Huang, "Over-the-air computation in MIMO multi-access channels: Beamforming and channel feedback," 2018. [Online]. Available: arxiv:1803.11129.

[44] G. Zhu, Y. Wang, and K. Huang, "Low-latency broadband analog aggregation for federated edge learning," 2018. [Online]. Available: arxiv:1812.11494.

[45] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[46] D. Liu, G. Zhu, J. Zhang, and K. Huang, "Wireless data acquisition for edge learning: Importance-aware retransmission," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Cannes, France, 2019, pp. 1–5.

[47] J.-H. Ahn, O. Simeone, and J. Kang, "Wireless federated distillation for distributed edge learning with heterogeneous data," 2019. [Online]. Available: arXiv.abs/1907.02745.

[48] X. Zhang, M. Peng, S. Yan, and Y. Sun, "Deep reinforcement learning based mode selection and resource allocation for cellular V2X communications," *IEEE Internet Things J.*, early access, Dec. 27, 2019, doi: 10.1109/JIOT.2019.2962715.

[49] X. Lu, X. Xiao, L. Xiao, C. Dai, M. Peng, and H. V. Poor, "Reinforcement learning-based microgrid energy trading with a reduced power plant schedule," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10728–10737, Dec. 2019.

[50] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "A graph neural network approach for scalable wireless power control," 2019. [Online]. Available: arXiv.abs/1907.08487.

[51] D. A. Temesgene, M. Miozzo, and P. Dini, "Dynamic control of functional splits for energy harvesting virtual small cells: A distributed reinforcement learning approach," 2019. [Online]. Available: arXiv.abs/1906.05735v1.

[52] Y. Chen, S. Deng, H. Zhao, Q. He, and H. G. Y. Li, "Data-intensive application deployment at edge: A deep reinforcement learning approach," in *Proc. IEEE Int. Conf. Web Services*, Jul. 2019, pp. 355–359.

[53] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[54] 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.

[55] X. Qiu, L. Liu, W. Chen, Z. Hong, and Z. Zheng, "Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8050–8062, Aug. 2019.

[56] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4005–4018, Jun. 2019.

[57] L. Huang, S. Bi, and Y. A. Zhang, "Deep reinforcement learning for online offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, early access, Jul. 24, 2019, doi: 10.1109/TMC.2019.2928811.

[58] L. Lei, X. Xiong, K. Zheng, W. Xiang, and X. Wang, "Multi-user resource control with deep reinforcement learning in IoT edge computing," 2019. [Online]. Available: arXiv:1906.07860.

[59] Q. Qi and Z. Ma, "Vehicular edge computing via deep reinforcement learning," 2019. [Online]. Available: arxiv:1901.04290.

[60] Z. Yang, Y. Liu, Y. Chen, and N. Al-Dhahir, "Cache-aided noma mobile edge computing: A reinforcement learning approach," 2019. [Online]. Available: arxiv:1906.08812.

[61] J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016. [Online]. Available: arxiv:1610.05492.

[62] K. Bonawitz *et al.*, "Practical secure aggregation for federated learning on user-held data," 2016. [Online]. Available: arxiv:1611.04482.

[63] M. Rusci, A. Capotondi, and L. Benini, "Memory-driven mixed low precision quantization for enabling deep network inference on micro-controllers," 2019. [Online]. Available: arxiv:1905.13082.

[64] P. S. Chandakkar, Y. Li, P. L. K. Ding, and B. Li, "Strategies for re-training a pruned neural network in an edge computing paradigm," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jun. 2017, pp. 244–247.

[65] U. Thakker, J. Beu, D. Gope, G. Dasika, and M. Mattina, "Run-time efficient RNN compression for inference on edge devices," in *Proc. 2nd Workshop Energy Efficient Mach. Learn. Cogn. Comput. Embedded Appl. (EMC2)*, Washington, DC, USA, 2019, pp. 26–30.

[66] R. Venkatesan and B. Li, "Diving deeper into MenTee networks," 2016. [Online]. Available: arxiv:1604.08220.

[67] I. Chakraborty, D. Roy, A. Ankit, and K. Roy, "Efficient hybrid network architectures for extremely quantized neural networks enabling intelligence at the edge," 2019. [Online]. Available: arxiv:1902.00460.

[68] I. Chakraborty, D. Roy, I. Garg, A. Ankit, and K. Roy, "PCA-driven hybrid network design for enabling intelligence at the edge," 2019. [Online]. Available: arxiv:1906.01493.

[69] A. Gupta *et al.*, "ProtoNN: Compressed and accurate KNN for resource-scarce devices," in *Proc. 34th Int. Conf. Mach. Learn.*, Feb. 2017, pp. 1331–1340.

[70] J. Hostetler, "Toward runtime-throttleable neural networks," 2019. [Online]. Available: arxiv:1905.13179.

[71] M. Blot, D. Picard, M. Cord, and N. Thome, "Gossip training for deep learning," 2016. [Online]. Available: arxiv:1611.09726.

[72] J. Daily, A. Vishnu, C. Siegel, T. Warfel, and V. Amatya, "Gossipgrad: Scalable deep learning using Gossip communication based asynchronous gradient descent," 2018. [Online]. Available: arxiv:1803.05880.

[73] H. Kim, J. Park, M. Bennis, and S. Kim, "On-device federated learning via blockchain and its latency analysis," 2018. [Online]. Available: arxiv:1808.03949.
[74] G. Li, L. Liu, X. Wang, X. Dong, P. Zhao, and X. Feng, "Auto-tuning neural network quantization framework for collaborative inference between the cloud and edge," in *Artificial Neural Networks and Machine Learning*, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds. Cham, Switzerland: Springer Int., 2018, pp. 402–411, doi: 10.1007/978-3-030-01418-6_40.
[75] L. Zhang, Z. Tan, J. Song, J. Chen, C. Bao, and K. Ma, "SCAN: A scalable neural networks framework towards compact and efficient models," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., New York, NY, USA: Curran Associates, Inc., 2019, pp. 4027–4036. [Online]. Available: http://papers.nips.cc/paper/8657-scan-a-scalable-neural-networks-framework-towards-compact-and-efficient-models.pdf
[76] M. J. Kusner, S. Tyree, K. Q. Weinberger, and K. Agrawal, "Stochastic neighbor compression," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 622–630.

Shuiguang Deng (Senior Member, IEEE) received the B.S. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 2002 and 2007, respectively.

He is currently a Full Professor with the First Affiliated Hospital, Zhejiang University School of Medicine, Hangzhou, as well as the College of Computer Science and Technology, Zhejiang University. He previously worked with the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2014 and Stanford University, Stanford, CA, USA, in 2015 as a Visiting Scholar. Up to now, he has published more than 100 papers in journals and refereed conferences. His research interests include edge computing, service computing, mobile computing, and business process management.

Prof. Deng was granted the Rising Star Award by IEEE TCSVC in 2018. He serves as an Associate Editor for IEEE ACCESS and *IET Cyber-Physical Systems: Theory & Applications*. He is a Fellow of IET.

Hailiang Zhao received the B.S. degree from the School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China, in 2019. He is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China.

His research interests include edge computing, service computing, and machine learning.

Dr. Zhao was a recipient of the Best Student Paper Award of IEEE ICWS 2019.

Weijia Fang received the master's degree in oncology and the Doctoral degree in oncology/surgery from Zhejiang University, Hangzhou, China, in 2005 and 2013, respectively.

He works with the Department of Medical Oncology, the First Affiliated Hospital, Zhejiang University School of Medicine, Hangzhou. He has authored and coauthored several original research publications in national and international peer-reviewed scientific and medical journals.

Jianwei Yin received the Ph.D. degree in computer science from Zhejiang University (ZJU), Hangzhou, China, in 2001.

He was a Visiting Scholar with the Georgia Institute of Technology, Atlanta, GA, USA. He is currently a Full Professor with the College of Computer Science, ZJU. Up to now, he has published more than 100 papers in top international journals and conferences. His current research interests include service computing and business process management.

Prof. Yin is an Associate Editor of the IEEE TRANSACTIONS ON SERVICES COMPUTING.

Schahram Dustdar (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in business informatics (Wirtschaftsinformatik) from the University of Linz, Linz, Austria, in 1990 and 1992, respectively.

He is a Full Professor of computer science (informatics) with a focus on Internet Technologies heading the Distributed Systems Group, Technische Universität Wien, Vienna, Austria. He has been the Chairman of the Informatics Section of the Academia Europaea since December 9, 2016. From 2004 to 2010, he was an Honorary Professor of information systems with the Department of Computing Science, University of Groningen, Groningen, The Netherlands. From December 2016 until January 2017, he was a Visiting Professor with the University of Sevilla, Seville, Spain, and from January until June 2017, he was a Visiting Professor with the University of California at Berkeley, Berkeley, CA, USA.

Prof. Dustdar was a recipient of the ACM Distinguished Scientist Award in 2009 and the IBM Faculty Award in 2012. He has been a member of the IEEE Conference Activities Committee since 2016, of the Section Committee of Informatics of the Academia Europaea since 2015, of the Academia Europaea: The Academy of Europe, Informatics Section since 2013. He is an Associate Editor of the IEEE TRANSACTIONS ON SERVICES COMPUTING, *ACM Transactions on the Web*, and *ACM Transactions on Internet Technology* and on the editorial board of IEEE INTERNET COMPUTING. He is the Editor-in-Chief of Computing (an SCI-ranked journal of Springer).

Albert Y. Zomaya (Fellow, IEEE) received the Ph.D. degree from the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U.K.

He is the Chair Professor of high-performance computing and networking with the School of Computer Science, University of Sydney, Sydney, NSW, Australia, where he also serves as the Director of the Centre for Distributed and High Performance Computing. He published more than 600 scientific papers and articles. He has authored, coauthored, or edited more than 30 books. His research interests are in the areas of parallel and distributed computing and complex systems.

Prof. Zomaya is the Editor-in-Chief of the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING and *ACM Computing Surveys* and serves as an associate editor for several leading journals. He served as an Editor-in-Chief for the IEEE TRANSACTIONS ON COMPUTERS from 2011 to 2014. He is a Chartered Engineer and a Fellow of AAAS and IET.