Edge Intelligence: the Confluence of Edge Computing and Artificial Intelligence

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

***Abstract*—Along with the deepening development in communication technologies and the surge of mobile devices, a brand-new computation paradigm, Edge Computing, is surging in popularity. Meanwhile, Artificial Intelligence (AI) applications are thriving with the breakthroughs in deep learning and the upgrade of hardware architectures. Billions of bytes of data, generated at the network edge, put great demands on data processing and structural optimization. Therefore, 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 a more optimal solution to the key concerns 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 edge. This article focuses on giving insights into this new inter-disciplinary field from a broader vision and perspective. It discusses the core concepts and the research road-map, which should provide the necessary background for potential future research programs in Edge Intelligence.**

***Index Terms*—Edge Intelligence, Edge Computing, Wireless Networking, Computation Offloading, Federated Learning.**

I. Introduction

COMMUNICATION technologies are undergoing a new revolution, i.e., the advent of the 5th generation cellular wireless systems (5G). 5G brings enhanced mobile broadband (eMBB), Ultra-Reliable Low Latency Communications (URLLC) and massive Machine Type Communications (mMTC). With the proliferation of the Internet of Things (IoTs), more and more data is created by widespread and geographically distributed mobile and IoT devices, instead of the mega-scale cloud datacenters [1]. Specifically, according to the prediction of Ericsson, 45% of the 40ZB global internet data will be generated by IoT devices in 2024 [2]. Offloading such huge data from the edge to cloud is intractable because it can lead to oppressive network congestion. Therefore, a more applicable way is handling user demands from edge directly, which leads to the birth of a brand-new computation paradigm, (Mobile $\rightarrow$ Multi-access) Edge Computing [3]. The subject of Edge Computing spans many concepts and technologies in diverse disciplines, including Service-oriented Computing (SOC), Software-defined Networking (SDN), Computer Architecture, etc. The principle of Edge Computing is to push the computation and communication resources from cloud to the edge of networks to provide services and perform computations, avoiding unnecessary communication latency and enabling faster responses for end users. Edge Computing is booming now.

No one can deny that Artificial Intelligence (AI) is developing unprecedentedly 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, 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 ramifications, i.e., Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and Generative Adversarial Networks (GANs), is the most advanced AI technology. Deep learning has made striking breakthroughs in a wide spectrum of fields, including computer vision, speech recognition, natural language processing, and board games. Besides, the hardware architectures and platforms keep upgrading with a rapid rate, which makes it possible to satisfy the requirements of the computation-intensive deep learning models. Application-specific accelerators are designed for further improvement in throughput and energy efficiency. In conclusion, driven by the breakthroughs in deep learning and the upgrade of hardware architectures, AI is undergoing sustained prosperity and development.

Considering that AI is functionally necessary for quickly analyzing huge volumes of data and extracting insights, there exists a strong demand to *integrate* Edge Computing and AI, which gives the birth of 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 complicated manner. Currently, the formal and internationally acknowledged definition of Edge Intelligence is non-existent. To deal with the problem, some researchers put forward their definitions. For example, Zhou et al. believe that the scope of Edge Intelligence should not be restricted to running AI models solely on the edge servers or devices but in the manner of the collaboration of edge and cloud [4]. They define six levels of Edge Intelligence, from *cloud-edge co-inference* (level 1) to *all on-device* (level 6). Zhang et al. define Edge Intelligence as the capability to enable edges to execute AI algorithms [5].

S. Deng, H. Zhao and J. Yin are with the College of Computer Science and Technology, Zhejiang University, 310058 Hangzhou, China. E-mail: {dengsg, hliangzhao, zjuyjw}@zju.edu.cn.

S. Dustdar is with the Distributed Systems Group, Technische Universitt Wien, 1040 Vienna, Austria. E-mail: dustdar@dsg.tuwien.ac.at.

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

In this paper, we propose to establish a broader vision and perspective. 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 the constrained optimization problems in Edge Computing with the help of popular and effective AI technologies. Here, AI is used for energizing edge with more intelligence and optimality. Therefore, it can be understood as *Intelligence-enabled Edge Computing (IEC)*.
2) *AI on edge* studies how to carry out the entire process of AI models on edge. It is a paradigm of running AI models training and inference 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 *Artificial Intelligence on Edge (AIE)*.

Edge Intelligence, currently in its early stage, is attracting more and more researchers and companies from all over the world. To disseminate the recent advances of Edge Intelligence, Zhou et al. have conducted a comprehensive and concrete survey of the recent research efforts on Edge Intelligence [4]. They survey the architectures, enabling technologies, systems, and frameworks from the perspective of AI models' training and inference. 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 this survey. Many concepts are still unclear and questions remain unsolved. The researching actuality motivates us to write this article to provide possible enlightening insights *with simple and clear classification*.

We commit ourselves to lucubrating Edge Intelligence in a broader vision and perspective. In Section II, we discuss the relation between Edge Computing and AI. In Section III, we demonstrate the research road-map of Edge Intelligence concisely with a hierarchical structure. Section IV and Section V elaborate the state of the art and grand challenges on *AI for edge* and *AI on edge*, respectively. Section VI concludes the article.

II. The 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. Edge Intelligence develops in the process of interaction and mutual promotion between Edge Computing and AI. On one hand, AI provides Edge Computing with technologies and methods, and Edge Computing is unleashing its potential at scale with AI; on the other hand, Edge Computing provides AI with scenarios and platforms, and AI broadly flourishes with Edge Computing.

AI provides Edge Computing with technologies and methods. In general, Edge Computing is a distributed computation paradigm, where software-defined networks have to be 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 competent to take on the task. Essentially, AI models extract unconstrained optimization problems from real scenarios and then find the asymptotically optimal solutions iteratively with Stochastic Gradient Descent (SGD) methods. Either statistical learning methods or deep learning methods can offer help and advice for the edge. Besides, reinforcement learning, including multi-armed bandit theory, multi-agent learning and deep Q-network (DQN), is playing a more and more 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 [6]. More and more data is 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 variety. 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 and more corporations participate in the design of chip architectures to support the edge computation paradigm and engage in the DNN acceleration on resource-limited IoT devices. The hardware upgrade on edge also injects vigor and vitality into AI.

III. Research road-map of Edge Intelligence

The architectural layers in the Edge Intelligence road-map, depicted in Fig. 1, describe a logical separation for the two directions respectively, i.e., *AI for edge* (left) and *AI on edge* (right). By 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 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 stays in the top of the road-map.

A. Quality of Experience

We believe that QoE should be *application-dependent* and determined by jointly considering multi-criteria from Performance, Cost, Privacy (Security), Efficiency and Reliability.

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*

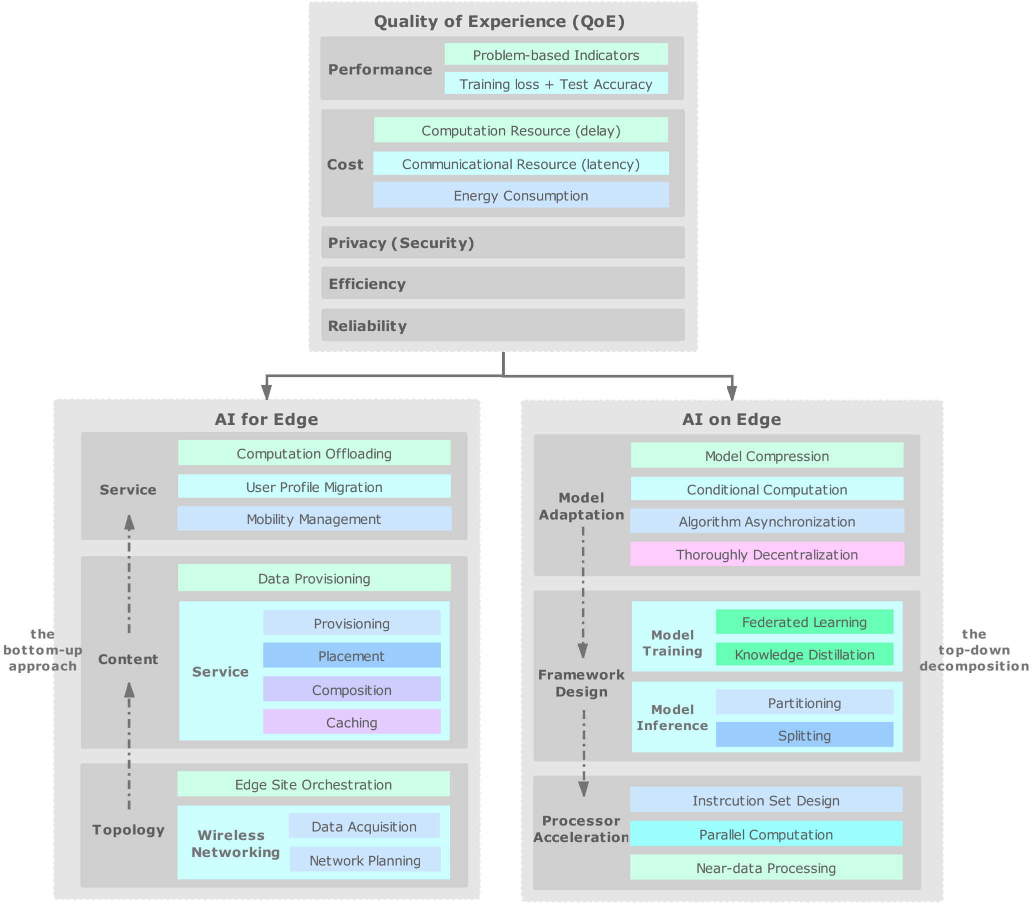

Fig. 1. The research road-map of Edge Intelligence.

when it comes into the computation offloading problems. It could be the service providers' *need-to-be-maximized revenue* and *need-to-be-minimized hiring costs* of Base Stations (BSs) when it comes into the service placement problems. As for the latter, performance is mainly constitutive of training loss and inference accuracy, which are the most important criteria for AI models. Although the computation scenarios have changed from cloud clusters to the synergetic 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, 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 [7]. 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 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 decentralized and synchronized way and the participated local users have a significant probability of failing to

complete the model upload and download due to the wireless network congestion.

B. A Recapitulation of IEC

The left of the road-map, depicted in Fig. 1, is *AI for edge*. We name this kind of work IEC (i.e. Intelligence-enabled Edge Computing) 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 data center with applications deployed, attached to a Small-cell Base Station (SBS). OES studies where to deploy and install wireless telecom equipment and servers. In recent years, research efforts on the management and automation of Unmanned Aerial Vehicles (UAVs) come into vogue [8] [9]. UAVs with a small server and an access point can be regarded as moving edge servers with strong maneuverability. Therefore, many works explore the scheduling and trajectory problems with the minimization of energy consumption of UAVs. 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, operating and managing. Fast data acquisition includes multiple access, radio resource allocation, and signal encoding/decoding. Network planning studies efficient management with protocols and middlewares. In recent years, there has been an increasing trend in *intelligent* networking. It builds an intelligent wireless communication mechanism by popular AI technologies. For example, Zhu et al. propose *Learning-driven Communication*, whose main principle is exploiting the coupling between communication and learning in edge learning systems [10].

For Content, we place 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 [11] [12]. The shared resources can also come from 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 study service placement from the perspective of Application Service Providers (ASPs). For example, [13] trys deploying services under the limited budget on basic communication and computation infrastructures. After that, multi-armed bandit 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 Quality of Experience (QoS) of mobile end users [14] [15] [16]. It opens research opportunities where 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 [17]. It opens research opportunities where multi-agent 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 and more research efforts focus on dynamically managing the radio and computational resources for multi-user multi-server edge computing systems, utilizing Lyapunov optimization technologies [18] [19]. In recent years, optimizing computation offloading decisions via DQN is popular [20] [21]. It models the computation offloading problem as a Markov decision process (MBP) and maximize 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 Deep Q-Network. 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. User profile migration is often associated with mobility management [22]. In [23], 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 increase 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. A Recapitulation of AIE

The right of the road-map is *AI on edge*. We name this kind of work AIE (i.e. Artificial Intelligence 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: Model Adaptation, Framework Design and 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 split and model split [24]. Data split can be further divided into master-device split, helper-device split and device-device split. The differences lie where the training samples come from and how the global model assembled and aggregated. Model split separates neural networks' layers and deploys them on different devices. It highly relies on sophisticated pipelines. Knowledge distillation-based frameworks may or may not be decentralized, and they rely on transfer learning technologies [25]. Knowledge distillation can enhance the accuracy of shallow student networks. It first trains a basic network on a basic dataset. After that, the learned features can be transferred to student networks to be trained on their datasets, respectively. The basic network can be trained on cloud or edge server 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* [7]. Federated Learning is proposed to preserve privacy when training the DNNs in a decentralized way. Without aggregating 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 Federated Learning thoroughly. For more details please refer to [7].

For Model Inference: Although model splitting is hard to realize for model training, it is a popular approach for model inference. To the best of our knowledge, model splitting/partitioning is the only approach that can be viewed as a *framework* for model inference. Other approaches such as model compression, input filtering, early-exit and so on can only be viewed as adaptations from existing frameworks, which will be introduced in the next paragraph and elaborated on carefully in Subsection V-A. A typical example on model inference on edge is [26], 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 *existing* training and inference frameworks, usually Federated Learning, to make them more applicable to the edge. Federated Learning has potential to be performed 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 the system performance. Methods to realize model adaptation include but not limited to Model Compression, Conditional Computation, Algorithm Asynchronization and Thoroughly 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 technologies such as Singular Value Decomposition (SVD), Huffman Coding, Principal Component Analysis (PCA) and some other acrobatics. 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 block-wise dropout [27]. Besides, random gossip communication can be utilized to reduce unnecessary calculations and model updates. Algorithm Asynchronization trys 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, (3) moving computation closer to memory (near-data processing), etc. The highly paralleled computing paradigms can be divided into temporal and spatial architectures [28]. The former architectures such as CPUs and GPUs can be accelerated by reducing the number of multiplications and increasing throughputs. The latter architectures can be accelerated by increasing data reuse with data flows. For example, [29] proposes an algorithm to accelerate CNN inference. The proposed algorithm converts a set of pre-trained weights into values under given precision. It also puts near-data processing into practice with an adaptive implementation of 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 launch a careful discussion on it. More details on hardware acceleration for DNN processing can be consulted in [28].

IV. AI FOR EDGE

In Subsection III-B, we divide the key concerns in Edge Computing into three categories: Topology, Content and Service. It just presents a classification and possible research directions but does not provide in-depth analysis on how to apply AI technologies to edge to generate more optimal solutions. This Section will remedy it. Fig. 2 gives an example of how AI technologies are utilized in the Mobile Edge Computing (MEC) environment. Firstly, 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. After that, the mathematical model should be constructed. At last, 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

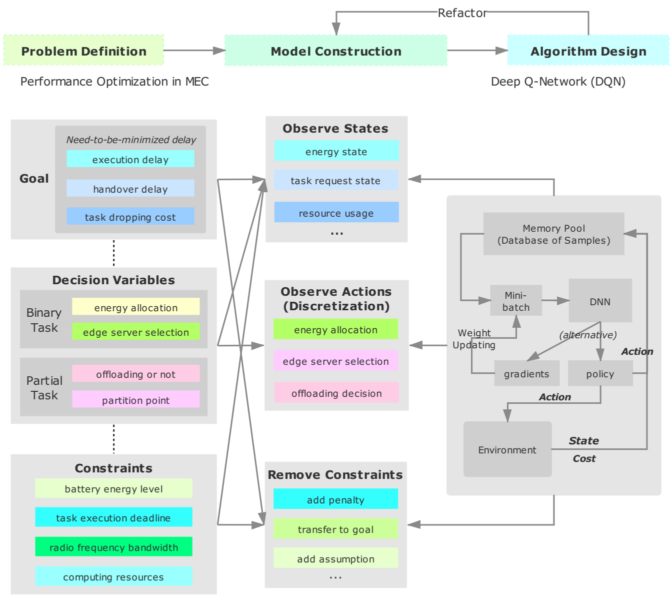

Fig. 2. The utilization of AI technology for performance optimization.

for example, we have to model the problem as a MDP with finite *states* and *actions*. Thus, the constraints can not exist in the long-term optimization problem. The most common way is transferring those constraints into penalty and adding the penalty to the optimization goal.

Considering that current research efforts on *AI for edge* concentrate on Wireless Networking, Service Placement, Service Caching and Computation Offloading, we only focus on these topics in the following Subsection. For research directions that haven't been touched and uncharted yet, we are looking forward to more exciting works.

A. State of the Art

1) Wireless Networking: 5G 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 [30]. We believe that AI has every right to be synergistically applied to realize intelligent network optimization in a fully online manner. One of the typical works is [10]. This paper 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 are in turn multiple access, radio resource management and signal encoding.

For learning-driven multiple access, it advocates that the unique characteristics of wireless channels should be exploited for functional computation. Over-the-air computation (AirComp) is a typical technique to realize it [31] [32]. [33] puts this principle into practice based on Broadband Analog Aggregation (BAA). Concretely, [33] suggests that the simultaneously transmitted model updates in Federated Learning should be analog aggregated by exploiting the waveform-superposition property of multi-access channels. The proposed BAA can dramatically reduce communication latency compared with traditional Orthogonal Frequency Division Multiple Access (OFDMA). [34] also explores the *over-the-air computation* for model aggregation in Federated Learning. Concretely, [34] puts the principle into practice by modeling the device selection and beamforming design as a sparse and low-rank optimization problem, which is highly intractably combinatorial. To solve the problem with fast convergence rate, this paper 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 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.

For learning-driven radio resource management, it advocates 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*. [35] puts the principle into practice. This paper proposed a retransmission protocol, named importance-aware automatic-repeat-request (importance ARQ). Importance ARQ makes the trade-off 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 channel noise.

For learning-driven signal encoding, it advocates that the signal encoding should be designed by jointly optimizing feature extraction, source coding, and channel encoding. A work puts this principle into practice is [36], 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 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 great research opportunities on the co-design of learning frameworks and data encoding.

Apart from *Learning-driven Communication*, some works contribute to *AI for Wireless Networking* from the perspective of power and energy consumption management. Shen et al. utilizes Graph Neural Networks (GNNs) to develop scalable methods for power control in K-user interference channels [37]. This paper first models the K-user interference channel as a complete graph, then it learns the optimal power control with a graph convolutional neural network. [38] studies 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 multi-agent learning, several distributed fuzzy Q-learning-based algorithms are tailored. This paper can be viewed as an attempt on coordination with broadcasting.

As we will expound later, Wireless Networking is often combined with Computation Offloading when it is studied in the form of optimization. State of the art of these works is listed in Subsection IV-A3.

2) Service Placement and Caching: Many researchers study service placement from the perspective of Application Service Providers (ASPs). They model the data and service (it could be composed and complicated) placement problem as a Markov Decision Process (MDP) and utilize AI technologies such as reinforcement learning to achieve the optimal placement decision. Typical work is [39]. This paper proposes a spatial-temporal algorithm based on Multi-armed bandit (MAB) and achieves the optimal placement decisions while learning the benefit. Concretely, it studies how many SBSs should be rent 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, a 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 attempts to integrate AI technologies with service placement is [40]. This work 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 DQN thoroughly. More details can be consulted in [41].

Service caching can be viewed as a complement to service placement. Edge servers can be equipped with special *service cache* to satisfy user demands on popular contents. A wide range of optimization problems on service caching are proposed to endow edge servers with learning capability. Sadeghi et al. study a sequential fetch-cache decision based on dynamic prices and user requests [17]. This paper 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 could be the hottest 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, AI technologies with strong optimization abilities have been extensively used in recent years. Among all these AI technologies, Q-learning and its derivate, DQN, are in the spotlight. For example, [42] designs a Q-learning-based algorithm for computation offloading. Concretely, it formulates the computation offloading problem as a non-cooperative game in multi-user multi-server edge computing systems and proves that Nash Equilibrium exists. Then, this paper proposes a model-free Q-learning-based offloading mechanism which helps 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 non-linear function approximation. For example, [20] studies the computation offloading for IoT devices with energy harvesting in multi-server 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 [21] [43], the computation offloading problem is formulated as a 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, [44] [45] jointly optimize task offloading decisions and wireless resource allocation to maximize the data acquisition and analysis capability of the network. [46] 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 sort of perspectives.

There also exist works who explore the task offloading problem with other AI technologies. For example, [47] proposes a long-short-term memory (LSTM) network to predict the task popularity and then formulates a joint optimization of the task offloading decisions, computation resource allocation and caching decisions. After that, a Bayesian learning automata-based multi-agent learning algorithm is proposed for optimality.

B. Grand Challenges

Although it is popular to apply AI technologies to edge for the generation of better solutions, there have many challenges to face. In the next several Subsections, we list grand challenges across the whole process of *AI for edge* research, which in turn are model establishment, algorithm deployment and the balance between optimality and efficiency. These challenges are closely related but each has its emphasis.

1) Model Establishment: If we want to use AI technologies, the built mathematical model has to be limited and the formulated optimization problem can not be unrestrained. On one hand, this is because the optimization basis of AI technologies, SGD (Stochastic Gradient Descent) and MBGD (Mini-Batch Gradient Descent) methods, may not work well if the original searching space is constrained. On the other hand, especially for MDPs, the state set and action set can not 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 and leads to the performance downgrade. It can be viewed as a compromise for the utilization of AI technologies. Therefore, how to establish appropriate system model faces 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 technologies and acrobatics. Actually, for *AI for edge*, the achieved solution mostly comes from iterative learning-based approaches. The status quo face great challenges when they 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 with optimality, the trade-off between optimality and efficiency can not be ignored when it comes to the resource-constrained edge. Thus, how to improve the usability and efficiency of edge computing system for different application scenarios with AI technologies embedded is a severe challenge. The trade-off 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 Subsection III-C, we divide the research efforts in *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 far limited 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 existing frameworks to make them more appropriate to edge, usually reducing resource occupation. As a result, Model Adaptation based on Federated Learning is prosperously developed. As we have mentioned before, Processor Acceleration will not be elaborated in details. Therefore, we only focus on Model Adaptation in the following Subsection. The state of the art is only discussed across Model Compression, Conditional Computation, Algorithm Asynchronization, and Thoroughly Decentralization.

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 Singular Value Decomposition (SVD), Huffman coding and Principal Component Analysis. This article will not launch a thorough introduction to them due to the 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 them.

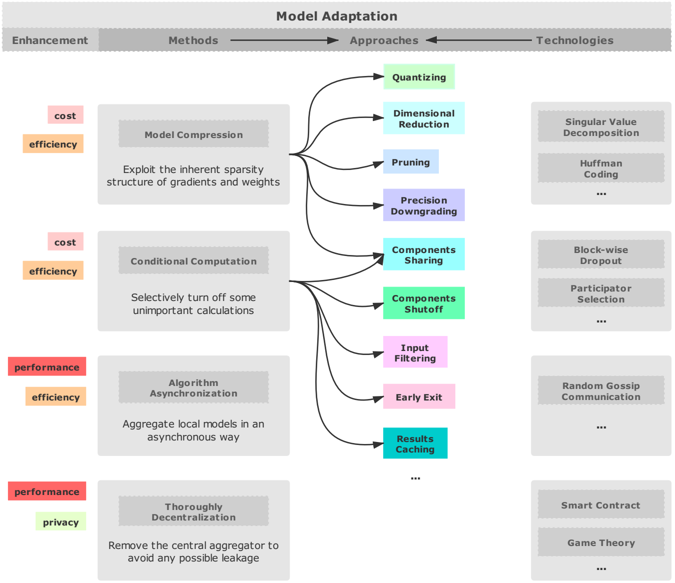

Fig. 3. Methods, approaches and technologies of Model Adaptation.

As we have mentioned before, communication efficiency is of the utmost importance for Federated Learning. Therefore, minimizing the number of rounds of communication is the principal goal when we move Federated Learning to the edge. A lot of works hammer at reducing the communication cost for Federated Learning from various perspectives. In [48], *structured updates* and *sketched updates* are proposed for reducing the uplink communication costs. For *structured updates*, the local update is learnt from a restricted lower-dimensional space; for *sketched updates*, the uploading model is compressed before sending to the central server. [49] designs a communication-efficient secure aggregation protocol for high-dimensional data. The protocol can tolerate up to 33.3% of participated devices failing to complete the protocol, i.e., the system is robust to the dropping out of participated users. [50] believes that DNNs are typically over-parameterized and their weights have significant redundancy. Meanwhile, pruning compensate with the lost performance. Thus, this paper 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. [51] exploits mixed low-bitwidth compression. It hammers at determining the minimum bit precision of each activation and weight under the given constraints on memory. [52] uses Binarized Neural Networks (BNNs), which have binary weights and activations to replace regular DNNs. This is a typical exploration of quantization. Analogously, [53] proposes hybrid network architectures combing binary and full-precision sections to achieve significant energy efficiency and memory compression with performance guaranteed. Thakker et al. study a compressed RNN cell implementation called Hybrid Matrix Decomposition (HMD) for model inference [54]. It divides the matrix of network weights into two parts: an unconstrained upper half and 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 acrobatic on compression, compared with traditional pruning or quantization. The numerical results show that it can not only achieve a faster run-time than pruning and but also retain more model accuracy than matrix factorization.

Some works also explore model compression based on partitioned DNNs. For example, [55] proposes an auto-tuning neural network quantization framework for collaborative inference between edge and cloud. Firstly, 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*. [56] 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 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, [57] proposes 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) [58], and then projects the entire data in a lower dimension with a sparse projection matrix. It jointly optimizes the projection and prototypes with explicit model size constraint. Chakraborty et al. proposes Hybrid-Net which has both binary and high-precision layers to reduce the degradation of learning performance [59]. Innovatively, this paper leverage 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 the hottest direction in *AI on edge* because it is easy to get started. However, the state-of-the-art works are usually not tied to specific application scenarios of edge computing systems. We are looking forward to exciting works set up on detailed 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 block-wise dropout [27]. 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, [60] instantiates 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 participated 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. propose GoSGD to train DNNs asynchronously [61]. 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. introduce GossipGraD, which can reduce the communication complexity greatly to ensure the fast convergence [62].

4) Thoroughly Decentralization: As demonstrated in Fig. 3, Thoroughly 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, blockchain technology and game-theoretical approaches can assist in total decentralization.

By leveraging blockchain, especially smart contract, 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 exists work who puts it into practice. In [63], the proposed blockchain-based federated learning architecture, BlockFL, takes edge nodes as miners. Miners exchange and verify all the local model updates contributed by each device and then run the Proof-of-Work (PoW). The miner who firstly completes the PoW generates a new block and receives the mining reward from the blockchain network. At last, each device updates its local model from the freshest block. In this paper, 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 is the beginning of everything. Firstly, a proper incentive mechanism may be necessary for data provisioning from 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 on the design of robust communication protocol. Therefore, there are huge challenges in terms of data availability.

2) Model Selection: At present, 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. Firstly, how to select the befitting threshold of learning accuracy and scale of AI models for quick deployment and delivery. Secondly, 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 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 heterogeneous edge devices are of great essence. A flexible coordination mechanism between cloud, edge, and device in both hardware and middlewares is imperative and urgently need to be designed. It opens research opportunities on a uniform API interface on edge learning for ubiquitous edge devices.

VI. Concluding Remarks

Edge Intelligence, although still in its primary stage, 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 Artificial Intelligence. 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 road-map. The former focuses on providing a better solution to the key concerns in Edge Computing with the help of popular and resultful 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 road-map 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 attempt to provide enlightening thought on the uncultivated land of Edge Intelligence. We hope this article can stimulate fruitful discussions on potential future research programs in Edge Intelligence.

References

[1] M. Asif-Ur-Rahman, F. Afsana, M. Mahmud, M. S. Kaiser, M. R. Ahmed, O. Kaiwartya, and A. James-Taylor, “Toward a heterogeneous mist, fog, and cloud-based framework for the internet of healthcare things,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4049–4062, June 2019.

[2] Ericsson, “Iot connections outlook: Nb-iot and cat-m technologies will account for close to 45 percent of cellular iot connections in 2024.” 2019. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/june-2019/iot-connections-outlook

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

[4] 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,” *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug 2019.

[5] X. Zhang, Y. Wang, S. Lu, L. Liu, L. Xu, and W. Shi, “Openei: An open framework for edge intelligence,” *CoRR*, vol. abs/1906.01864, 2019. [Online]. Available: http://arxiv.org/abs/1906.01864

[6] 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 of Things Journal*, vol. 4, no. 5, pp. 1125–1142, Oct 2017.

[7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *AISTATS*, 2016.

[8] J. Xu, Y. Zeng, and R. Zhang, “Uav-enabled wireless power transfer: Trajectory design and energy optimization,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5092–5106, Aug 2018.

[9] B. Li, Z. Fei, and Y. Zhang, “Uav communications for 5g and beyond: Recent advances and future trends,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241–2263, April 2019.

[10] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, “Towards an intelligent edge: Wireless communication meets machine learning,” *CoRR*, vol. abs/1809.00343, 2018. [Online]. Available: http://arxiv.org/abs/1809.00343

[11] H. Wu, S. Deng, W. Li, J. Yin, Q. Yang, Z. Wu, and A. Y. Zomaya, “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: Springer International Publishing, 2017, pp. 625–640.

[12] 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. 54 258–54 269, 2018.

[13] L. Chen, J. Xu, S. Ren, and P. Zhou, “Spatiotemporal edge service placement: A bandit learning approach,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 8388–8401, Dec 2018.

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

[15] S. Deng, L. Huang, J. Taheri, J. Yin, M. Zhou, and A. Y. Zomaya, “Mobility-aware service composition in mobile communities,” *IEEE Trans. Systems, Man, and Cybernetics: Systems*, vol. 47, no. 3, pp. 555–568, 2017. [Online]. Available: https://doi.org/10.1109/TSMC.2016.2521736

[16] 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 Computing*, vol. 9, no. 5, pp. 664–671, 2016. [Online]. Available: https://doi.org/10.1109/TSC.2015.2418765

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

[18] 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 *2018 IEEE International Conference on Ubiquitous Intelligence Computing*, Oct 2018, pp. 671–678.

[19] 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 *2019 IEEE International Conference on Web Services*, Jul 2019, pp. 193–200.

[20] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for iot devices with energy harvesting," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1930–1941, Feb 2019.

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

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

[23] C. Zhang, H. Zhao, and S. Deng, "A density-based offloading strategy for iot devices in edge computing systems," *IEEE Access*, vol. 6, pp. 73 520–73 530, 2018.

[24] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *CoRR*, vol. abs/1812.02858, 2018. [Online]. Available: http://arxiv.org/abs/1812.02858

[25] 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," *CoRR*, vol. abs/1809.03428, 2018. [Online]. Available: http://arxiv.org/abs/1809.03428

[26] E. Li, Z. Zhou, and X. Chen, "Edge intelligence: On-demand deep learning model co-inference with device-edge synergy," in *Proceedings of the 2018 Workshop on Mobile Edge Communications, MECOMM@SIGCOMM 2018, Budapest, Hungary, August 20, 2018*, 2018, pp. 31–36.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

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

[29] J. Lee, J. K. Eshraghian, K. Cho, and K. Eshraghian, "Adaptive precision cnn accelerator using radix-x parallel connected memristor crossbars," *ArXiv*, vol. abs/1906.09395, 2019.

[30] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.

[31] O. Abari, H. Rahul, and D. Katabi, "Over-the-air function computation in sensor networks," *CoRR*, vol. abs/1612.02307, 2016. [Online]. Available: http://arxiv.org/abs/1612.02307

[32] G. Zhu, L. Chen, and K. Huang, "Over-the-air computation in MIMO multi-access channels: Beamforming and channel feedback," *CoRR*, vol. abs/1803.11129, 2018. [Online]. Available: http://arxiv.org/abs/1803.11129

[33] G. Zhu, Y. Wang, and K. Huang, "Low-latency broadband analog aggregation for federated edge learning," *CoRR*, vol. abs/1812.11494, 2018. [Online]. Available: http://arxiv.org/abs/1812.11494

[34] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *CoRR*, vol. abs/1812.11750, 2018. [Online]. Available: http://arxiv.org/abs/1812.11750

[35] D. Liu, G. Zhu, J. Zhang, and K. Huang, "Wireless data acquisition for edge learning: Importance aware retransmission," *CoRR*, vol. abs/1812.02030, 2018. [Online]. Available: http://arxiv.org/abs/1812.02030

[36] J.-H. Ahn, O. Simeone, and J. Kang, "Wireless federated distillation for distributed edge learning with heterogeneous data," *ArXiv*, vol. abs/1907.02745, 2019.

[37] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "A graph neural network approach for scalable wireless power control," *ArXiv*, vol. abs/1907.08487, 2019.

[38] D. A. Temesgene, M. Miozzo, and P. Dini, "Dynamic control of functional splits for energy harvesting virtual small cells: a distributed reinforcement learning approach," *ArXiv*, vol. abs/1906.05735v1, 2019.

[39] L. Chen, J. Xu, S. Ren, and P. Zhou, "Spatiotemporal edge service placement: A bandit learning approach," *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 8388–8401, Dec 2018.

[40] 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 *2019 IEEE International Conference on Web Services*, Jul 2019, pp. 355–359.

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

[42] 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 Transactions on Vehicular Technology*, pp. 1–1, 2019.

[43] 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 of Things Journal*, vol. 6, no. 3, pp. 4005–4018, June 2019.

[44] L. Huang, S. Bi, and Y. A. Zhang, "Deep reinforcement learning for online offloading in wireless powered mobile-edge computing networks," *CoRR*, vol. abs/1808.01977, 2018. [Online]. Available: http://arxiv.org/abs/1808.01977

[45] L. Lei, H. X. andXiong Xiong, K. Zheng, W. Xiang, and X. Wang, "Multi-user resource control with deep reinforcement learning in iot edge computing," *ArXiv*, vol. abs/1906.07860, 2019.

[46] Q. Qi and Z. Ma, "Vehicular edge computing via deep reinforcement learning," *CoRR*, vol. abs/1901.04290, 2019. [Online]. Available: http://arxiv.org/abs/1901.04290

[47] Z. Yang, Y. Liu, Y. Chen, and N. Al-Dhahir, "Cache-aided noma mobile edge computing: A reinforcement learning approach," *ArXiv*, vol. abs/1906.08812, 2019.

[48] J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *CoRR*, vol. abs/1610.05492, 2016. [Online]. Available: http://arxiv.org/abs/1610.05492

[49] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for federated learning on user-held data," *CoRR*, vol. abs/1611.04482, 2016. [Online]. Available: http://arxiv.org/abs/1611.04482

[50] 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 *2017 IEEE International Conference on Edge Computing (EDGE)*, June 2017, pp. 244–247.

[51] M. Rusci, A. Capotondi, and L. Benini, "Memory-driven mixed low precision quantization for enabling deep network inference on microcontrollers," *CoRR*, vol. abs/1905.13082, 2019. [Online]. Available: http://arxiv.org/abs/1905.13082

[52] R. Venkatesan and B. Li, "Diving deeper into mentee networks," *CoRR*, vol. abs/1604.08220, 2016. [Online]. Available: http://arxiv.org/abs/1604.08220

[53] I. Chakraborty, D. Roy, A. Ankit, and K. Roy, "Efficient hybrid network architectures for extremely quantized neural networks enabling intelligence at the edge," *CoRR*, vol. abs/1902.00460, 2019. [Online]. Available: http://arxiv.org/abs/1902.00460

[54] U. Thakker, J. G. Beu, D. Gope, G. Dasika, and M. Mattina, "Run-time efficient RNN compression for inference on edge devices," *CoRR*, vol. abs/1906.04886, 2019. [Online]. Available: http://arxiv.org/abs/1906.04886

[55] 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," *CoRR*, vol. abs/1812.06426, 2018. [Online]. Available: http://arxiv.org/abs/1812.06426

[56] L. Zhang, Z. Tan, J. Song, J. Chen, C. Bao, and K. Ma, "SCAN: A scalable neural networks framework towards compact and efficient models," *CoRR*, vol. abs/1906.03951, 2019. [Online]. Available: http://arxiv.org/abs/1906.03951

[57] A. Gupta, C. G. S. Suggala, A. Gupta, H. Simhadri, B. Paranjape, A. Kumar, S. Goyal, R. Udupa, M. Varma, and P. Jain, "Protonn: Compressed and accurate knn for resource-scarce devices," in *Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70*, February 2017.

[58] M. J. Kusner, S. Tyree, K. Q. Weinberger, and K. Agrawal, "Stochastic neighbor compression," in *ICML*, 2014.

[59] I. Chakraborty, D. Roy, I. Garg, A. Ankit, and K. Roy, "Pca-driven hybrid network design for enabling intelligence at the edge," *CoRR*, vol. abs/1906.01493, 2019. [Online]. Available: http://arxiv.org/abs/1906.01493

[60] J. Hostetler, "Toward runtime-throttleable neural networks," *CoRR*, vol. abs/1905.13179, 2019. [Online]. Available: http://arxiv.org/abs/1905.13179

[61] M. Blot, D. Picard, M. Cord, and N. Thome, "Gossip training for deep learning," *CoRR*, vol. abs/1611.09726, 2016. [Online]. Available: http://arxiv.org/abs/1611.09726

[62] J. Daily, A. Vishnu, C. Siegel, T. Warfel, and V. Amatya, “Gossipgrad: Scalable deep learning using gossip communication based asynchronous gradient descent,” *CoRR*, vol. abs/1803.05880, 2018. [Online]. Available: http://arxiv.org/abs/1803.05880

[63] H. Kim, J. Park, M. Bennis, and S. Kim, “On-device federated learning via blockchain and its latency analysis,” *CoRR*, vol. abs/1808.03949, 2018. [Online]. Available: http://arxiv.org/abs/1808.03949

Shuiguang Deng is currently a full professor at the College of Computer Science and Technology in Zhejiang University, China, where he received a BS and PhD degree both in Computer Science in 2002 and 2007, respectively. He previously worked at the Massachusetts Institute of Technology in 2014 and Stanford University in 2015 as a visiting scholar. His research interests include Edge Computing, Service Computing, Mobile Computing, and Business Process Management. He serves as the associate editor for the journal IEEE Access and IET Cyber-Physical Systems: Theory & Applications. Up to now, he has published more than 100 papers in journals and refereed conferences. In 2018, he was granted the Rising Star Award by IEEE TCSVC. He is a fellow of IET and a senior member of IEEE.

Hailiang Zhao received the B.S. degree in 2019 from the school of computer science and technology, Wuhan University of Technology, Wuhan, China. He is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. He has been a recipient of the Best Student Paper Award of IEEE ICWS 2019. His research interests include edge computing, service computing and machine learning.

Jianwei Yin received the Ph.D. degree in computer science from Zhejiang University (ZJU) in 2001. He was a Visiting Scholar with the Georgia Institute of Technology. He is currently a Full Professor with the College of Computer Science, ZJU. Up to now, he has published morethan 100 papers in top international journals and conferences. His current research interests include service computing and business process management. He is an Associate Editor of the IEEE Transactions on Services Computing.

Schahram Dustdar is a Full Professor of Computer Science (Informatics) with a focus on Internet Technologies heading the Distributed Systems Group at the TU Wien. He is Chairman of the Informatics Section of the Academia Europaea (since December 9, 2016). He is elevated to IEEE Fellow (since January 2016). From 2004-2010 he was Honorary Professor of Information Systems at the Department of Computing Science at the University of Groningen (RuG), The Netherlands.

From December 2016 until January 2017 he was a Visiting Professor at the University of Sevilla, Spain and from January until June 2017 he was a Visiting Professor at UC Berkeley, USA. He is a member of the IEEE Conference Activities Committee (CAC) (since 2016), of the Section Committee of Informatics of the Academia Europaea (since 2015), a member of the Academia Europaea: The Academy of Europe, Informatics Section (since 2013). He is recipient of the ACM Distinguished Scientist award (2009) and the IBM Faculty Award (2012). He is an Associate Editor of 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 is the Chair Professor of High Performance Computing & Networking in the School of Information Technologies, University of Sydney, and he also serves as the Director of the Centre for Distributed and High Performance Computing. Professor Zomaya published more than 600 scientific papers and articles and is author, co-author or editor of more than 20 books. He is the Founding Editor in Chief of the IEEE Transactions on Sustainable Computing and serves as an associate editor for more than 20 leading journals. Professor Zomaya served as an Editor in Chief for the IEEE Transactions on Computers (2011-2014). He is a Chartered Engineer, a Fellow of AAAS, IEEE, and IET. Professor Zomayas research interests are in the areas of parallel and distributed computing and complex systems.