Intelligence-Empowered Mobile Edge Computing: Framework, Issues, Implementation, and Outlook

Kai Jiang, Chuan Sun, Huan Zhou, Xiuhua Li, Mianxiong Dong, and Victor C. M. Leung

Abstract

Recently, artificial intelligence (AI) is undergoing a sustained success renaissance as it can substantially improve networks' cognitive performance and intelligence, thereby contributing to fully unleashing the potential of big data. Pushing the AI frontiers to the network edge in this context and trends has given rise to an emerging interdiscipline, namely, edge intelligence (EI). Indeed, EI can sink the cloud's processing capabilities to the edge side, and provide real-time response while enabling more intelligent services with high performance. However, the successful realization of EI is still in its infancy. Thus, this article aims to provide a comprehensive study of this young field from a broader perspective. We first discuss the prior knowledge based on which we take a holistic overview of EI, including its key concepts, advantages, and development trend. Then we highlight the collaboration modes in EI, and discuss two typical case categories. Subsequently, the entire processes of model training and inference in EI are elaborated. Finally, we discuss a typical application scenario and its specific embodiment of EI and strive to shed light on some potential challenges, which may facilitate the transformation of EI from theory to practice.

Introduction

With the popularity of mobile devices and the advances of wireless access technologies in fifth generation (5G) networks, flourishing mobile applications have led to explosive growth of data traffic [1]. According to the International Data Corporation, global data center traffic will reach 163 ZB by 2025, of which more than 75 percent of the data will be processed at the edge of the network [2, 3].

These emerging applications generally have intense requirements on quality of service (QoS), which makes mobile cloud computing (MCC) infeasible as the long transmission distance may contribute to extra cost, delay, and unreliability. Indeed, cloud servers are spatially far from end devices. The accompanying challenges driven by the transmission distance in cloud-based processing have attracted widespread academic concern. Despite the continuous efforts spent on enhancing the channel bandwidth, the utilization efficiency of the radio spectrum is notably reaching its theoretical bound [4]. Thus, such efforts will not be sufficient, and a fundamental innovation that breaks the bottleneck of massive task processing in 5G networks is urgently required.

All these challenges have accelerated the development of mobile edge computing (MEC). Specifically, MEC is a promising solution to cope with innovative applications by its physical proximity to the information-generation sources. Compared to MCC, MEC pushes computation, caching, and network functions toward the network edges to perform task processing and provide services, avoiding unnecessary transmission delay [5, 6]. Such patterns not only accommodate the expansion requirements of the computation capabilities of end devices, but also promise several benefits, including reduced energy consumption, lower delay, privacy protection, more accessibility, and context awareness [7, 8].

At the same time, the development of artificial intelligence (AI) has experienced a fall, a leap, and again a fall, until spectacular leaps were made in the past decade. Specifically, driven by breakthroughs in hardware upgrading and a series of neural networks, AI expands technological innovations with its superiority for data analysis and extracting insights, especially in time-varying and complex networks [9, 10]. Pushing the AI frontiers to the network edge under this context has given rise to an emerging interdiscipline, namely, edge intelligence (EI). Notably, EI is the next development stage of MEC. It is not the simple integration of MEC and AI, but the complementation and mutual benefits. Instead of entirely relying on the cloud, EI can sink the cloud's processing capabilities at the edge, and provide real-time response while enabling more intelligent services with high performance. Furthermore, the leakage and loss of users' private data can also explicitly be alleviated by EI.

It is undeniable that although EI has attracted a tremendous amount of interest in recent years, the successful implementation of EI is still in its infancy. This article aims to provide a comprehensive study in this young field from a broader perspective. We first review the prior knowledge, based on which we take a holistic overview of EI, including its key concepts and advantages, and the development trend. Then we highlight the collaboration modes in EI, and discuss two typical case categories. Subsequently, the processes of

Digital Object Identifier:
10.1109/MNET.101.2100054

Kai Jiang and Huan Zhou (corresponding author) are with China Three Gorges University; Chuan Sun and Xiuhua Li are with Chongqing University; Mianxiong Dong is with Muroran Institute of Technology; Victor C. M. Leung is with Shenzhen University.

model training and inference in EI are elaborated holistically. Finally, we discuss a typical application scenario and its specific embodiment of EI, and as strive to shed light on some potential challenges, which may facilitate the transformation of EI from theory to practice.

Some Prior Knowledge of EI

This section introduces the prior knowledge based on which we take a holistic overview of EI, including its basic concepts, development trend, and motivation.

Basic Concept of EI

EI is currently considered to be an eye-catching emerging technology trend, which has risen through the combination of MEC and AI. More generally, EI can be understood as intelligence-empowered edge computing. It combines the advantages of both MEC and AI, in which they can complement each other and be mutually beneficial. On one hand, AI provides MEC with technologies and methods, and MEC can unleash its potential and scalability with AI. On the other hand, MEC provides AI with scenarios and platforms, and AI can expand its applicability with MEC.

Essentially, EI can enable edge equipment to perform model training and inference locally, avoiding frequent communication with the cloud platform. The emergence of EI is highly nontrivial due to the concerns of system efficiency, scheduling optimization, and privacy protection in MEC. At present, EI's main research directions include edge-cloud collaboration, model segmentation, reduction of redundant data transmission, and design of lightweight acceleration architecture. Furthermore, from a technical perspective, AI models can extract insights from practical edge environments and seek high-quality asymptotic solutions iteratively. The methods represented by deep learning and reinforcement learning (RL) have gradually become the most popular AI techniques in EI. Deep learning can automatically extract features and detect edge anomalies, while RL, including multi-agent RL [11] and deep reinforcement learning (DRL), which refers to the process of realizing objectives via multiple steps and suitable decisions, is playing a growing important role in the real-time decision making of edge networks.

Development Trend of EI

It has already been six years since the related concepts of EI were first proposed. In the meantime, although both academia and industry have made great efforts in EI fields, its successful implementation is still a long way off. This is because there is still a lack of standardized ideas on EI, which significantly hinders its promotion and development.

We take a step back and discuss the essential evolutionary events of EI. Specifically, the European Telecommunications Standard Institute (ETSI) is at the forefront of EI-related exploration, and published a technical white paper "NO.11 Mobile Edge Computing — A key technology towards 5G" in September 2015. In this paper, the proposal and technical requirements of EI were formally proposed for the first time [12]. The corresponding referenced platform, challenges, and typical application scenarios of EI were discussed.

In MEC Build Congress 2017, Microsoft proposed the intelligent cloud and intelligent edge concepts, which will assist the strategic planning and decision making of cloud-edge collaboration. Meanwhile, Alibaba Cloud published the Link Edge platform in March 2018, aiming to sink the cloud's processing capabilities to the edge side, and identify which AI-based methods may be viable in the edge network. In August 2018, Alibaba and some Internet of Things (IoT) industry chain partners launched an IoT Connectivity Alliance (ICA) group to promote EI's technical standardization development. They delivered several specifications for technique requirements and formally published a technical white paper on EI. In the latter part of 2018, the Edge Computing Consortium (ECC) hosted the MEC Industry Summit in Beijing with EI. All these efforts have further accelerated the development of EI applications and ultimately expanded the market size. Overall, EI still demands constant efforts in all aspects. Not only academic theories, but also more realistic challenges and requirements need to be considered.

Why Do We Need EI?

The integration of MEC and AI is an inevitable trend. They promote each other in the following aspects.

Lower Latency and Bandwidth Consumption: In the conventional MCC, the end users' data are stored and processed at the remote cloud data center, which will undoubtedly consume many bandwidth resources and bring tremendous pressure on the network. EI has become a reliable solution to the above challenges. By moving the processing capability from the cloud to the edge, AI services are deployed near mobile users, and the cloud does not need to participate in the whole process of service. Hence, EI can achieve lower latency and reduce bandwidth consumption.

Adapting to Time-Varying Environments: As the middle layer, edge nodes reduce the pressure of cloud infrastructure due to the access of massive end devices. However, the management of such a complex edge network architecture involving bandwidth, computing power, storage capacity, and so on is a significant problem. Since conventional network optimization methods need to rely on fixed and complex mathematical knowledge, they cannot adapt to the rapidly changing network environment. The emergence of EI is expected to solve this problem. Due to its powerful learning ability, AI technology enables the edge to tackle sophisticated network optimization problems (e.g., task offloading, user association, resource allocation) and make adaptive decisions by extracting valuable information from data.

Richer Edge Application Scenarios: Dispersed end devices under different application scenarios continuously generate massive heterogeneous data (e.g., text, image, and video). Edge servers cannot handle such a large amount of data due to computing resources and storage capacity limitations. In such cases, AI can provide fast analysis and diversified inferences to improve the QoS of services. For example, DRL enables networks to identify patterns and extract features from the

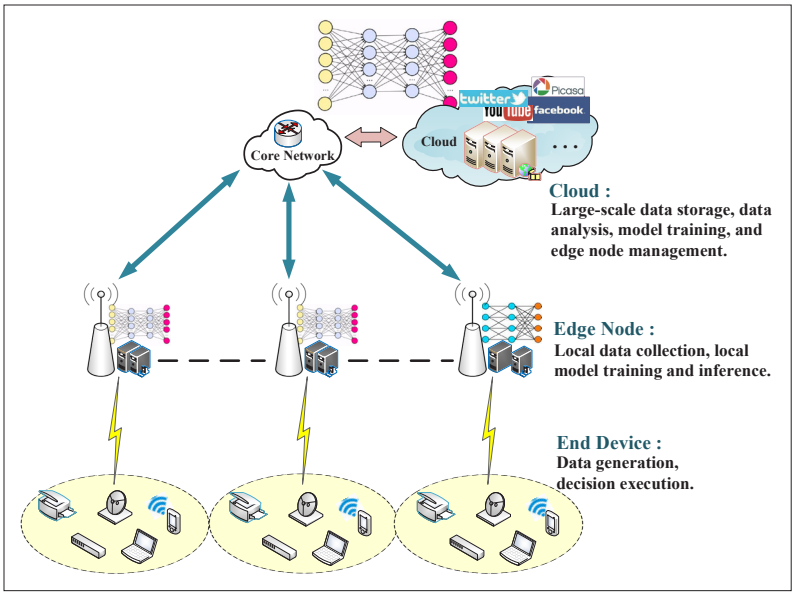

FIGURE 1. The general hierarchical architecture with cloud-edge-end collaboration.

data automatically. By feeding data into the DRL model, the edge sides can also update decisions and increase efficiency. Most importantly, these rich data will improve the AI technology's performance. Specifically, to improve the accuracy of the AI model and improve the model structure, a large number of data are needed for training. The data collected at the edge side can explicitly provide sufficient training resources for AI models. Therefore, AI and MEC are complementary and mutually beneficial.

MEC Is an Enabler for Ubiquitous AI: AI facilitates our daily lives dramatically and has achieved great success in many fields. As its application areas expand continuously, the ultimate goal of its development is AI everywhere. For this goal, AI should be "closer" to users. MEC has more advantages than MCC to achieve this. First, the edge server is in closer proximity to the data source. Second, MEC is more economical and accessible for mobile operators. Finally, with the connection of IoT devices, MEC is able to provide more abundant AI application scenarios. Hence, MEC is a crucial enabler for ubiquitous AI.

MEC Can Be Popularized with AI Applications: In the fledgeling stage of MEC, improving network performance by MEC has been investigated all along, and its dominant application is also the focus of research. Recently, emerging real-time applications have fully exhibited MEC's advantages, and played an essential role in the development of MEC. Since these applications require high computation, bandwidth resources, and ultra-low delay, MEC meets these intense requirements efficiently. With the emergence of more and more AI application scenarios related to IoT, MEC with privacy protection and ultra-low latency will continue to show its advantages.

Collaboration Scope and Case Category

In this section, we highlight the collaboration modes in EI, and discuss two typical case categories.

Collaboration Scope

Stemming from the trend in recent years, EI is evolving from providing single intelligent services into a more powerful cooperative operation stage. Many studies indicate that the scope of EI should not be restricted solely to one organizational level. Thus, cloud-edge-end coordination is integrated into the EI paradigm's design to fully exploit the resources across various hierarchies. However, how to connect and coordinate available resources among the hierarchical architectures so as to optimize the training model and make trade-offs between multi-criteria is still of great essence. Indeed, AI can implement the compatibility and coordination between the heterogeneous resources of different levels via data offloading. Figure 1 shows the general hierarchical architecture with cloud-edge-end collaboration in the EI paradigm, which can be further divided into cloud-edge collaboration, edge-edge collaboration, and edge-end collaboration, respectively.

Cloud-Edge Collaboration: Recently, cloud-edge collaboration has grown as a relatively mature collaboration mode and has already attracted widespread investigation in academia and industry. Among them, the cloud is responsible for analyzing and mining big data, as well as model training and upgrading. Its powerful storage and computing capacity can ensure large-scale and long-term intelligent processing. In comparison, the edge is responsible for data computation and caching within the local scope. Its real-time performance can support local and short-cycle intelligent decision making and execution well. This collaboration enables edges to further offload workloads to the cloud for processing, and supports the controlled and orderly flow of data between the edge and the cloud to complete the closed loop of autonomous learning, which are of substantial significance for fully utilizing the edge and cloud resources.

Edge-Edge Collaboration: Considering the cost constraints, one edge node may suffer from inherent resource limitation in processing extensive applications separately. To improve the resource utilization in the system, the edge nodes in a cluster form are enabled to communicate and compute with each other rather than work individually. Thus, the synergies between various edge nodes should be harnessed to facilitate good system performance in edge-edge collaboration mode. It is worth noting that dynamic resource requirements, heterogeneous subsystem conditions, and different connections among edge nodes often exist in edge-edge collaboration. The capacities of an independent edge node may also be time-varying. Therefore, it is also crucial that efficient cooperative edge scheduling strategies in the system are available.

Edge-End Collaboration: The term "end" in edge-end collaboration refers to end devices, typically represented by a series of IoT devices, mobile phones, and vehicles. Edge-end collaboration is a lightweight model to relieve the increasing conflict between task requests' diversity and edge equipment's single competency, which effectively enhances edge nodes' processing capacities. Moreover, the high correlation between end devices and specific application

Case categories	Methods used	Features	Benefits
Information prediction	Deep learning	Heterogeneous data processing; automatic feature extraction	Improve the prediction accuracy and range
Strategy optimization	Deep learning Reinforcement learning	Required to calculate labels; consider complex constraints systematically; interact with unknown environments	Real-time decision making; autonomic network management; complex problems; self-adaptation

TABLE 1. Typical case categories of EI.

scenarios makes edge-end collaboration more focused on intelligent scheduling and secure access of end devices. In the edge-end collaboration, end devices collect the real-time aware data and offload them to the edge server. Then the edge server performs centralized calculation and analysis for the data from multiple sources, and sends operation identifiers to the end devices for providing end users with reliable services. Overall, due to the close relationship between the end devices and the users, edge-end collaboration is considered as an essential step toward the implementation of EI applications.

Typical Case Categories

Stemming from the recent studies, we can categorize the related applications of EI into two categories: information prediction and strategy optimization [13, 14].

Information Prediction: The core concept of information prediction is to predict the evolution of network conditions by leveraging context information in order to facilitate good performance in a dynamic system. Generally, to improve an EI system's performance, the information that needs to be predicted includes the uncertain parameters related to the dynamic wireless channel and network conditions (e.g., channel state information, resources occupation rate, power control), and the private information related to the end users (location-based information, social-based behavior, content popularity, etc.). According to practical predicted information, the system can adapt to time-varying environments and further promote the resource utilization for various task requirements. For instance, harnessing the predicted content popularity in each time slot can significantly optimize the cache replacement strategy, which can alleviate redundant traffic and reduce the content delivery delay in edge caching.

Currently, the most popular method used in information prediction is deep learning, which can be further divided into supervised learning and semi-supervised learning, as shown in Table 1. Specifically, deep learning can process vast heterogeneous data and perform automatic feature extraction. Compared to conventional model-based prediction methods, deep-learning-based prediction methods can effectively improve the prediction accuracy and range.

Strategy Optimization: Besides designing system strategies or resource allocation beforehand via information prediction, we can also use some adaptive methods and learn to optimize the strategy online while providing more efficient and intelligent services. Unlike conventional data-driven methods, optimizing real-time strategies is promising for tackling the complex NP-hard problem in time-varying networks. Moreover, under autonomic network management, the methods of strategy optimization can capture the environment's hidden dynamics well by enhancing the intelligence of edge networks.

Correspondingly, the leading method used in strategy optimization is RL, including DRL, in which deep neural networks (DNNs) are adopted to avoid the curse of dimensionality. Specifically, RL enables the agent to adaptively learn the optimal strategy through repeated interaction within a specific context in discrete time steps.

Implementation of EI

In this section, we elaborate all of the processes of model training and inference in EI, and introduce some related performance indicators.

Model Training

The convergence of MEC and AI relies on the efficient distributed model training and inference along the edge-cloud continuum, which is critical for enabling high-quality EI service deployment. In general, we classify the architectures into three modes based on the deployment location of model training, namely, centralized, decentralized, and hybrid, respectively.

Centralized Mode: In the centralized training mode, the trained model is deployed on the cloud computing platform, from which the data preprocessing, model training, and message brokering are mainly performed by the cloud. Specifically, the training model is implemented through cloud-edge collaboration, and its performance heavily depends on the quality of network connections. In the training phase, the edge node collects the data within the local scope and uploads it to the cloud in real time; these data are generated by applications, sensors, devices, video channels, and social media of end users. After analyzing and caching the edge node's data, the model is continually trained in centralized training clusters using aggregated data. Notably, although the centralized mode promises the potential for searching the system's optimal solution, model training's complexity grows exponentially with global network state data. Meanwhile, as the deployed model on cloud computing platforms is spatially far from users, the user data must pass through multiple networks in a wide area network. Data delivery and unpredictable network connections may cause prohibitive transmission delay and overhead. Furthermore, this mode is also more vulnerable due to user data resources' high concentricity, which explicitly leads to leakage and loss of sensitive private data.

Decentralized Mode: Nowadays, flourishing mobile applications, especially latency-sensitive applications, always have intense requirements on ultra-reliable and low-latency communications, which may make centralized training no longer feasible for today's developments. Indeed, each edge node's inferences could be significantly different among different local states. Meanwhile, users expect to benefit from EI while keeping their privacy. Thus, an appropriate way to avoid

Model training	Model inference	Cloud (central node)	Edge node
Centralized	Centralized	Training + Inference	\
Centralized (sharing model)	Decentralized	Training	Inference
Decentralized	Decentralized	\	Training + Inference

TABLE 2. Model training and inference modes.

this dilemma is to train the model in a decentralized manner. In the decentralized mode, there is no centralized node (cloud), and all the edge nodes perform equal roles. Remarkably, all the edge nodes have a neural network mark, implying that each of them can perform model training independently. Each edge node trains its model locally with local data, which preserves private information locally. However, the training process on an independent edge node makes it easy to perform overfitting due to the correlation among the restricted data source, while the inferences of different edge nodes are usually mutually influential in the system. Thus, to obtain the global training model by sharing local training improvement, multiple edge nodes have to undertake model training or data analysis synergistically, and the training set is generated by themselves [15]. In this way, the global model can be trained without the cloud computing platform's intervention. Furthermore, emerging federated learning is applied to the data- sensitive area under the decentralized training mode. It is worth noticing that conventional decentralized training focuses on consuming data at the edge side, while federated learning focuses more on privacy protection.

Hybrid Mode: Indeed, it is unrealistic for each edge node to train a comprehensive model on its own, considering latency and consumption. Thus, the common way adopted in EI architecture is a hybrid mode that combines the centralized and decentralized modes. Note that the hybrid mode is not limited in practical deployment and is flexible to adapt to the application scenario. In this mode, the edge nodes may train the DRL model by either decentralized updates with each other or centralized training with the cloud computing platform. Specifically, each edge node trains partial parameters and aggregates them to a central node for the global model upgrade. The private data is only gathered in the edge node, resulting in privacy preservation, weaker than the decentralized mode but more robust than the centralized mode.

In addition, the main ideas of reducing training complexity are divided into system-level and method-level at present. The system level is devoted to finding suitable training methods, while the method level tends to formulate superior models or introduce prior knowledge.

Model Inference

Model inference (i.e., running the trained model) happens after training. Efficient model inference is equally critical to the implementation of EI. According to the trained model category described above, model inference can be executed either on the cloud or on lone edge nodes. There are three typical model training and inference modes, as shown in Table 2.

Specifically, the typical inference modes also include centralized and decentralized. In the former, the model training and inference are both finished on the cloud, and the inference results will be sent to each edge node separately. While in the latter, each edge node can perform its own model inference locally. Notably, in the centralized training mode, the cloud can either maintain one training model for a centralized inference of all edge nodes or send the trained sharing model to the edge node for distributed inferences. More detail of the implementation process is shown in Fig. 2. Generally, methods including supervised learning, unsupervised learning, and single-agent RL are commonly used for centralized inference, while multi-agent RL is the standard method for decentralized inference.

Performance Indicators

We can elaborate the critical performance indicators in an EI system, which are related to the model training and model inference, respectively. We describe them as follows [10].

Training Loss and Inference Accuracy: Training loss and inference accuracy are the most important criteria for AI models. As for the former, training loss usually occurs in the model training stage, and has common objectives with convergence. The loss function is the bias between the predicted value and the absolute value, representing the matching degree between the training model and the original data. Therefore, it is expected to minimize the loss function by selecting appropriate training methods and samples. As for the latter, accuracy refers to the correct predictions from the total number of input samples to the model inference, which directly influences the validity of the trained model. For the service with high reliability requirements, accuracy is essential, which greatly affects the quality of user experience. For example, in the vehicle edge networks, the accuracy of the decision making affects the driving comfort of vehicles and the driving safety of drivers and passengers.

Energy Efficiency: For EI, energy efficiency is mainly affected by the sample data size, training model complexity, and edge equipment resources. Considering the hardware constraints in decentralized edge nodes, the training model must be energy-efficient. Therefore, the pursuit of efficiency is a critical factor for improving the existing algorithms and models, especially for EI. Furthermore, since an economic EI system should consider the trade-off between QoS and the energy consumption caused by the model inference, a reasonable model inference process should efficiently reduce the services' energy consumption or delay. Notably, in order to improve the efficiency of training and inference, methods such as model compression, conditional calculation, and algorithm synchronization are generally proposed.

Fairness: The EI system not only needs to consider system performance and energy efficiency, but also fairness among various edge nodes. Load balancing needs to be considered to achieve reasonable utilization of resources among the cooperative edge nodes. Also, with more and more users accessing the EI services, various resources, such as bandwidth, computation, and caching resources, are occupied simultaneously. Thus, resource conflict is also a factor that needs to be

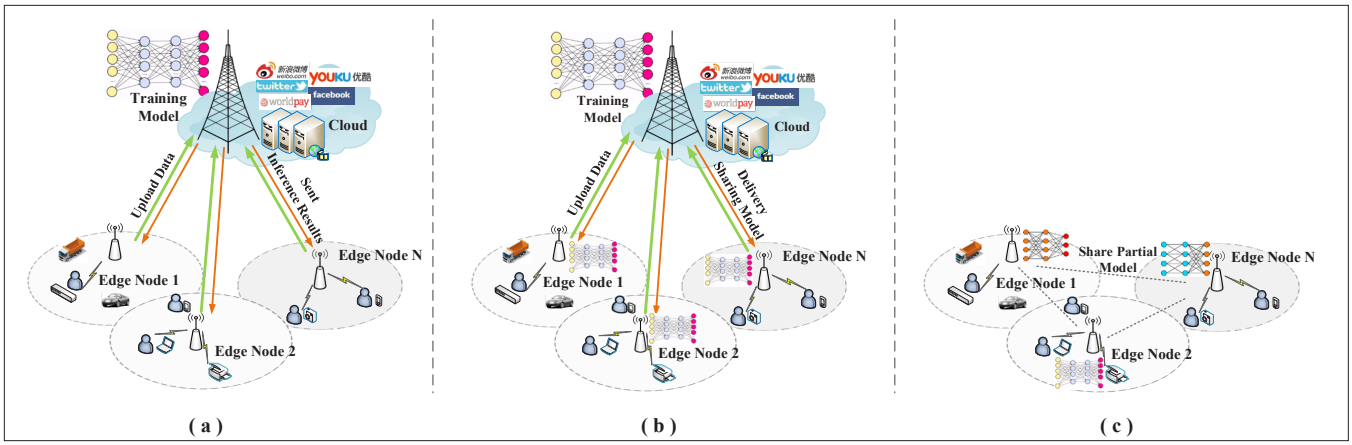

FIGURE 2. The process of model training and inference: a) centralized training, centralized inference; b) centralized training, decentralized inference; c) decentralized training, decentralized inference.

resolved in model training and inference reasonably.

Security and Privacy: As cases of data leakage are commonplace, privacy protection has become a hot topic. During the model training stage, the end devices' original data will carry much personal privacy information. Therefore, the processing of these data is crucial. Meanwhile, model training may exhibit unintended behavior during the training process. The trained model may be subject to adversarial attacks, which should be quantified via robustness indicators. Hence, it is also essential to protect the privacy and data security near the data source for an EI application during the model inference stage. Since wireless eavesdroppers may eavesdrop on computing tasks, it may be not safe to offload tasks through wireless channels. The transmission of application data can be protected by encrypting the user side and decrypting it on the target server side. However, this may increase the transmission delay and execution delay, thereby reducing system performance. Furthermore, sharing the same storage and computing resources among multiple mobile users will lead to leakage and loss of private data.

Case Study, Open Challenges, and Future Directions of EI

EI is an emerging interdisciplinary field with many challenges as well as tremendous opportunities. In this section, we first present our view of a typical application scenario and then strive to shed light on some potential challenges, which may facilitate the transformation of EI from theory to practice.

Case Study of EI in the Internet of Vehicles

We have already witnessed that the key value propositions of EI are exemplified in some applications. In this part, a typical application scenario of EI is discussed, together with its specific embodiment.

Along with the recent advances in wireless access technologies and the steady growth of vehicles on the road, more and more vehicles are able to connect to roadside units (RSUs), which has paved a path toward data-driven intelligent transportation. Specifically, AI can substantially improve the cognitive performance and intelligence of the Internet of Vehicles (IoVs) to adapt to rapidly changing dynamic environments, and provide multiple task requirements for resource allocation, computing task scheduling, and vehicle trajectory prediction. On this basis, EI has exhibited fascinating potential for handling various intelligent vehicle applications by adding AI services to edge RSUs.

Now, we discuss a specific embodiment of EI in IoVs, as shown in Fig. 3. Specifically, we consider an IoV architecture with a macro base station (MBS), K RSUs, and U vehicles, for which each RSU is endowed with MEC servers for computing offloading, and edge caching. The vehicle requests various contents frequently, and F is the total amount of all available contents. Furthermore, we consider vehicle-to-vehicle communication in the system, where each vehicle with processing capacities is considered to be an edge node. Hence, under cloud-edge collaboration, the requesting vehicle can concurrently offload its tasks and download the requested content with the connected RSUs, vehicles, or remote MBS.

We expect to investigate the optimization of edge computing and caching, where decisions of caching, computing, and resource allocation are considered jointly. Meanwhile, the system conditions, resource capabilities, offloading and caching states, and vehicular mobility intensity are uncertain and time-varying. Under the current states, we aim to determine the subset of the nearby edge nodes (RSUs, vehicles) and their resources to respond to the requesting vehicle. In view of the dynamic variants and the enormous action-state space in practical vehicular edge networks, we exploit a double deep Q-network (DDQN)-based method, which can efficiently determine the optimal action according to a tremendous amount of input data.

In the following, we give a brief description of DDQN. DDQN is an improvement of the traditional DQN algorithm, and its basic idea is to separate the selection and evaluation of actions by constructing different action functions. The DDQN-based method can approximately obtain the optimal Q value by using the updated DNN parameter q. Meanwhile, the agent stores an experience tuple that includes the current state, selected action, reward, and next state into the experience replay buffer at each time slot to break

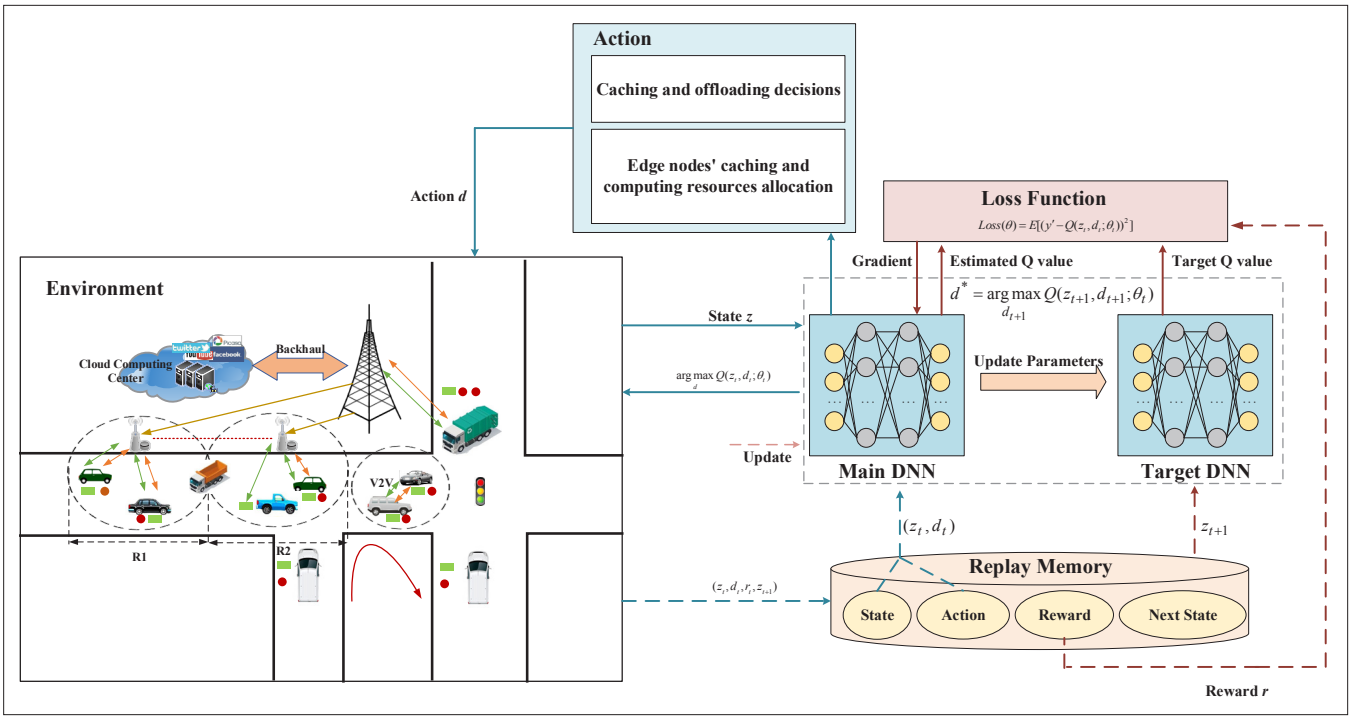

FIGURE 3. The process of DDQN to solve the optimization of edge computing and caching in IoV.

the correlation among data training. Furthermore, two neural networks with the same structure but different parameters are maintained to disrupt the pertinence. The target neural network is used to acquire the temporal difference target Q value, while the main neural network is used to evaluate the Q function. The stored experience tuples can be randomly sampled for training the main network and the target network. Thus, the input of the main network is the current state from the vehicular edge network environment, the training state, and the training action from the replay buffer. The output of the main network is the action that is adopted by the MBS, RSUs, and vehicles. Note that the weight parameters $\hat{\theta}_j$ of the target DNN are updated periodically by the counterpart θ_j of the main DNN by $\hat{\theta} = \zeta\theta + (1 - \zeta)\,\hat{\theta}$ with $\zeta \ll 1$.

In this DDQN model, the system state consists of the available RSU and vehicles, the available caches, the caching and computing resources at the RSU and vehicles, the transmission channel information, and the vehicles' mobility. Accordingly, for system action, the agent has to decide whether/where the requested content should be cached or the computation task be offloaded, and how many coded packets should be cached. Hence, the system action consists of the RSUs' and vehicles' caching and computing resources for the requesting vehicle. Finally, as the value of the reward is negatively correlated to the value of the objective function, we aim to minimize the cost of communication, storage, and computation.

The system operation is as follows. At first, the system collects the state and sends it to the main network and replay buffer. Then, based on the current state and experience tuples, the agent uses the main network and target network to determine the next action. Specifically, we have to pre-process the offloading, caching decision, and resource allocation of the system with random policy for a sufficiently long time. Then the main neural network and target neural network update their weight parameters θ_j and $\hat{\theta}_j$ based on the samples from the replay buffer, respectively. After that, the fixed Q-target is used to generate the target Q value $\acute{y}$ as $\acute{y}_j = r_j + \varepsilon Q(z_j + 1, \text{argmax}_{dj+1} Q(z_{j+1}, d_{j+1}\,;\theta_j)\,;\hat{\theta}_j)$ and transmit it to the main network. After receiving $\acute{y}$, the main network trains the Q value toward the target value by minimizing the loss function. Based on the Q value and experience tuples, the agent makes gradient guiding updates and then generates the next action, while stochastic gradient descent is performed until the Q value converges to the optimal value.

In addition, the BS, RSUs, and vehicles execute computing offloading and content caching based on the determined action. The environment feeds back an immediate reward to the agent based on the next new state. If the current action satisfies all constraints of the proposed joint optimization, and the system utility of the current policy is greater than the existing maximal system utility, the immediate reward is updated, and the environment updates its state based on the current policy. However, if constraints are violated in the proposed problem, the agent receives a punitive negative directly. For long-term consideration, there is a cumulative discounted system utility in DDQN. When the cumulative discounted system utility converges, the optimal policy of computation offloading, content caching, and resource allocation are successfully trained.

Open Challenges

Although EI has brought us great benefits, there are still some challenges that need to be reconsidered before it can be deployed commercially.

System Dynamics and Openness: EI needs to consider the dynamic conditions of wireless networks and the openness of wireless channels.

First, the wireless network environment is not stable; users joining and leaving the network, sending new service requests, and users' movement will lead to dynamic changes of the network. EI needs real-time decision making and requires online learning to respond to the dynamic changes of the network promptly, which puts forward strict requirements on the training time and cost of the AI model. Second, interference and fading in wireless channels inevitably reduce communication quality, which affects the learning accuracy and convergence speed of EI. Therefore, EI needs to be robust in various uncertain environments.

Hardware and Network Architecture Support: End devices' steady growth will drive the vigorous development of EI applications on distributed edge networks. These computation-intensive and delay-sensitive EI applications always require massive resources to operate model training and inference. Therefore, in view of the time-varying and uncertain system conditions in practical edge networks, an advanced network architecture is desired to relieve the training and inference pressure under limited resources. Furthermore, designing an advanced network architecture can also ensure ultra-reliable and low-latency communications in EI via efficient resource management and task scheduling of edge nodes.

Lightweight Training Models: As existing AI models often enable advanced functions via complex network structures, it is necessary to discuss how to deploy and execute these AI workloads on edge nodes with limited resources. We should design a more lightweight AI model to adapt to the resource constraints on edge nodes. Compression techniques such as Early Exit of Inference (EEoI), knowledge distillation, quantization, and weight pruning are effective methods to realize lightweight AI models. However, although model compression can help AI models run at the edge, it is often accompanied by the loss of model accuracy. The static compression technique cannot adapt to the dynamic hardware configuration and loads of edge nodes. Therefore, it is a potential research direction to apply the emotional compression technique to edge nodes with complicated conditions.

Security and Privacy: As cases of data leakage are commonplace, privacy protection has become a hot topic recently. When training the model, the uploaded data may carry a lot of personal privacy information. Therefore, the processing of private data is crucial. Although many techniques (e.g., federated learning) have been proposed to solve these issues, the raw data can still be inferred and reconstructed. There are two main approaches to protect the security and privacy of EI. The first is to add noise to protect the original data, but the noise will inevitably interfere with the reasoning in the cloud. The second method is homomorphic encryption, which encrypts data, and the final inference result is still ciphertext. Meanwhile, the data of end devices can be considered to be processed locally.

Conclusion

In this article, we conduct a comprehensive study of the recent research in EI from a broader perspective. Specifically, we first discuss the prior knowledge, based on which we provide a holistic overview of EI, including its key concepts, advantages, and development trend. Then we highlight the collaboration modes in EI, and discuss two typical case categories. Subsequently, all of he processes of model training and inference in EI are elaborated. Finally, we discuss a typical application scenario and its specific embodiment of EI, as well as strive to shed light on some potential challenges, which may facilitate the transformation of EI from theory to practice.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (62172255 and 61872221) and JSPS KAKENHI (JP20F20080).

References

[1] Q. Pham *et al.*, "A Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art," *IEEE Access*, vol. 8, 2020, pp. 116,974–117,017.
[2] H. Zhou *et al.*, "DRAIM: A Novel Delay-Constraint and Reverse Auction-Based Incentive Mechanism for WiFi Offloading," *IEEE JSAC*, vol. 38, no. 4, 2020, pp. 711–22.
[3] K. Zhang *et al.*, "Mobile Edge Computing and Networking for Green and Low-Latency Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 5, May 2018, pp. 39–45.
[4] Y. Dai *et al.*, "Artificial Intelligence Empowered Edge Computing and Caching for Internet of Vehicles," *IEEE Wireless Commun.*, vol. 26, no. 3, June 2019, pp. 12–18.
[5] N. Abbas *et al.*, "Mobile Edge Computing: A Survey," *IEEE IoT J.*, vol. 5, no. 1, Feb. 2018, pp. 450–65.
[6] X. Wang *et al.*, "Cache in the Air: Exploiting Content Caching and Delivery Techniques for 5G Systems," *IEEE Commun. Mag.*, vol. 52, no. 2, Feb. 2014, pp. 131–39.
[7] D. Zeng *et al.*, "Resource Management at the Network Edge: A Deep Reinforcement Learning Approach," *IEEE Network*, vol. 33, no. 3, May/June 2019, pp. 26–33.
[8] X. Li *et al.*, "Hierarchical Edge Caching in Device-to-Device Aided Mobile Networks: Modeling, Optimization, and Design," *IEEE JSAC*, vol. 36, no. 8, Aug. 2018, pp. 1768–85.
[9] C. Wang *et al.*, "Artificial Intelligence Enabled Wireless Networking for 5G and Beyond: Recent Advances and Future Challenges," *IEEE Wireless Commun.*, vol. 27, no. 1, Feb. 2019, pp. 16–23.
[10] S. Deng *et al.*, "Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence," *IEEE IoT J.*, vol. 7, no. 8, Aug. 2020, pp. 7457–69.
[11] F. Wang *et al.*, "Intelligent Video Caching at Network Edge: A Multi-Agent Deep Reinforcement Learning Approach," *Proc. IEEE INFOCOM*, 2020, pp. 2499–2508.
[12] S. Wang *et al.*, "A Survey on Mobile Edge Networks: Convergence of Computing Caching and Communications," *IEEE Access*, vol. 5, 2017, pp. 6757–79.
[13] N. C. Luong *et al.*, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *IEEE Commun. Surveys & Tutorials*, vol. 21, no. 4, 2019, pp. 3133–74.
[14] H. Zhang *et al.*, "Power Control Based on Deep Reinforcement Learning for Spectrum Sharing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, 2020, pp. 4209–19.
[15] Z. Zhou *et al.*, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence with Edge Computing," *Proc. IEEE*, vol. 107, no. 8, 2019, pp. 1738–62.

Biographies

Kai Jiang [S'21] (jiangkai1217@whu.edu.cn) received his M. S. degree from China Three Gorges University in 2021. He is currently pursuing a Ph. D. degree in cyberspace security at Wuhan University, China. His research interests include mobile edge computing, reinforcement learning, and the Internet of Vehicles.

Chuan Sun [S'20] (c.sun@cqu.edu.cn) is a Ph. D. student at Chongqing University, China. He received his B. S. degree from Wuhan University of Science and Technology, China, in 2017. His current research interests include multi-access edge computing, recommender systems, and machine learning.

Huan Zhou [M'14] (zhouhuan117@gmail.com) received his Ph.D. degree from the Department of Control Science and Engineering at Zhejiang University. He was a visiting scholar at Temple University from November 2012 to May 2013 and

a CSC supported postdoctoral fellow at the University of British Columbia, Vancouver, Canada, from November 2016 to November 2017. Currently, he is a full professor at the College of Computer and Information Technology, China Three Gorges University. He was a Lead Guest Editor of *Pervasive and Mobile Computing*, TPC Chair of EAI BDTA 2020, Local Arrangement Chair of I-SPAN 2018, Special Session Chair of the 3rd Int'l. Conf. Internet of Vehicles (IOV 2016), and a TPC member of IEEE GLOBECOM, ICC, ICCCN, and so on. He has published more than 60 research papers in international journals and conferences, including *IEEE JSAC*, *TPDS*, *TWC*, and so on. His research interests include opportunistic mobile networks, VANETs, mobile data offloading, and mobile edge computing. He received the Best Paper Awards of I-SPAN 2014 and I-SPAN 2018, and is currently serving as an Associate Editor for *IEEE Access* and the *EURASIP Journal on Wireless Communications and Networking*.

XIUHUA LI [S'12, M'19] (lixiuhua1988@gmail.com) received his B.S. degree from the Honors School, Harbin Institute of Technology, China, in 2011, his M.S. degree from the School of Electronics and Information Engineering, Harbin Institute of Technology, in 2013, and his Ph.D. degree from the Department of Electrical and Computer Engineering, University of British Columbia in 2018. He joined Chongqing University through the One-Hundred Talents Plan of Chongqing University in 2019. He is currently a tenure-track assistant professor with the School of Big Data & Software Engineering, Chongqing University, and is also the Dean of the Institute of Intelligent Software and Services Computing associated with the Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Education Ministry, China. His current research interests are 5G/B5G mobile Internet, mobile edge computing and caching, big data analytics, and machine learning.

MIANXIONG DONG (mx.dong@csse.muroran-it.ac.jp) received B.S., M.S., and Ph.D. degrees in computer science and engineering from the University of Aizu, Japan. He is the vice president and youngest ever professor at Muroran Institute of Technology, Japan. He was the recipient of the IEEE TCSC Early Career Award 2016, the IEEE SCSTC Outstanding Young Researcher Award 2017, the 12th IEEE ComSoc Asia-Pacific Young Researcher Award 2017, the Funai Research Award 2018, and NISTEP Researcher 2018 (one of only 11 people in Japan) in recognition of significant contributions in science and technology by MEXT, Japan. He is a Clarivate Ana lytics 2019 Highly Cited Researcher (Web of Science).

VICTOR C. M. LEUNG [S'75, M'89, SM'97, F'03] (vleung@ieee.org) received B.S. (Hons.) and Ph.D. degrees, both in electrical engineering, from the University of British Columbia, where he holds the positions of professor and TELUS Mobility Research Chair in the Department of Electrical and Computer Engineering. He is a Fellow of the Royal Society of Canada, the Engineering Institute of Canada, and the Canadian Academy of Engineering. He has contributed more than 700 technical papers, 26 book chapters, and 5 books in the areas of wireless networks and mobile systems. He was a Distinguished Lecturer of IEEE Communications Society. He has served or is serving on the Editorial Boards of *IEEE Transactions on Computers*, *Wireless Communications*, *Vehicular Technology*, *IEEE Wireless Communications Letters*, and several other journals, and has contributed to the Organizing and Technical Program Committees of numerous conferences. He was a winner of the 2012 UBC Killam Research Prize and the IEEE Vancouver Section Centennial Award.