Intelligent Offloading in Multi-Access Edge Computing: A State-of-the-Art Review and Framework

Bin Cao, Long Zhang, Yun Li, Daquan Feng, and Wei Cao

The authors introduce the basic concept of MEC and its main applications. They review existing fundamental works using various ML-based approaches, and they discuss potential issues for AI in MEC in future work.

Abstract

Multi-access edge computing (MEC), which is deployed in the proximity area of the mobile user side as a supplement to the traditional remote cloud center, has been regarded as a promising technique for 5G heterogeneous networks. With the assistance of MEC, mobile users can access computing resource effectively. Also, congestion in the core network can be alleviated by offloading. To adapt in stochastic and constantly varying environments, augmented intelligence (AI) is introduced in MEC for intelligent decision making. For this reason, several recent works have focused on intelligent offloading in MEC to harvest its potential benefits. Therefore, machine learning (ML)-based approaches, including reinforcement learning, supervised/unsupervised learning, deep learning, as well as deep reinforcement learning for AI in MEC have become hot topics. However, many technical challenges still remain to be addressed for AI in MEC. In this article, the basic concept of MEC and main applications are introduced, and existing fundamental works using various ML-based approaches are reviewed. Furthermore, some potential issues of AI in MEC for future work are discussed.

Introduction

Toward the fifth generation (5G) [1] mobile communications network, ubiquitous and intelligent cloud computing is one of the key technologies. However, the powerful cloud center is usually deployed far away from mobile users, and thus huge amounts of traffic are usually transmitted through multiple intermediate nodes. As a result, heavy load, congestion, delay, energy consumption, and so on could be incurred, and these would weaken the advantages of cloud computing. Therefore, multi-access edge computing (MEC) [2], which moves computing resource from the core network to the edge, is proposed as a natural design.

Figure 1 illustrates the typical architecture and main applications of MEC in heterogeneous networks (HetNets) [3]. Different from the remote cloud center, MEC is a distributed network architecture at the edge network. From the perspective of the MEC operator, various applications could run on the edge network closer to mobile users, thereby reducing congestion in the core network. From the perspective of the mobile user, a resource-constrained device would be liberated from computation-intensive and delay-sensitive applications to enhance the user's quality of service (QoS). First, the processing task could be assigned from a computation-limited individual device to available resource-rich MEC servers through an uplink channel. Next, the corresponding computing resource should be allocated for processing on the MEC side. After processing is finished, the computation result would be fed back to the mobile user through a downlink channel. This procedure is widely called "offloading" [3]. In an MEC-enabled system, such as MEC-enabled vehicular networks and the Internet of Things (IoT), the following key issues should be addressed regarding offloading:

- The cooperation and competition among mobile users and MEC operators
- The resource matching for communication and computation
- The practical mechanism design facing the characteristics of mobility, stochasticity, and heterogeneity

Moreover, since offloading is a "connection" between the mobile user and cloud provider, it is more or less related to most of the key problems in MEC (i.e., offloading decision, resource allocation, mobility management, content caching, green energy supply, security, and privacy [3]). Therefore, offloading plays a critical role in MEC naturally, and the rest of this article mainly focuses on offloading in MEC.

In the practical MEC system, the offloading decision making optimization is sophisticated because the influence factors are multi-dimensional, randomly uncertain, and time varying. As a result, traditional approaches (e.g., game theory, optimization theory) meet the obstacles and limitations in such complex scenarios. As is well known, artificial intelligence based on machine learning (ML) [4] can extract useful information from massive data, learning and providing various functions for optimization, prediction, and decision in a stochastic environment, such as analysis for service characteristics using data mining technology. On one hand, artificial intelligence usually

Digital Object Identifier:
10.1109/MCOM.2019.1800608

Bin Cao, Long Zhang, and Yun Li (corresponding author) are with the School of Communication and Information Engineering, Chongqing University of Post and Telecommunications; Daquan Feng is with Shenzhen University; Wei Cao is with the University of Electronic Science and Technology of China.

relies on massive data, and this is beneficial for artifical intelligence applied in MEC. On the other hand, due to the caused high training and learning costs in artificial intelligence, MEC can provide rich computing resource and low-latency service for intelligent computing. Introducing intelligence computation, MEC is expected to make the edge network realize self-optimization and self-adaptation, thereby forming an augmented intelligence (AI) decision making system in MEC. Specifically, AI in MEC does not just apply artificial intelligence to design protocols or algorithms; it is a systematic framework to construct a smart MEC decision making system with the aid of artificial intelligence.

Intelligent Approaches for Offloading in MEC

In the following, some mainstream ML-based approaches in offloading are briefly introduced. To better understand ML approaches, the illustrations for reinforcement learning, supervised and unsupervised learning, deep learning, and deep reinforcement learning are shown in Fig. 2.

Reinforcement Learning in Offloading

Reinforcement learning, inspired by behavioral psychology, is concerned with how a decision maker ought to choose the optimal action by continuously interacting with the system environment [5]. The goal of reinforcement learning is to select an action for each state of the system so as to maximize cumulative reward in the long term (delayed reward instead of immediate reward). Reinforcement learning is suitable for automatic control and decision making issues under a stochastic and dynamic environment. Different from other categories of ML, reinforcement learning cannot learn from data; instead, it has to learn from its own experience.

Markov Decision Process (MDP): A reinforcement learning environment is usually formulated as an MDP that provides a mathematical framework used to model decision making. More specifically, a decision maker chooses an action on state s_t. Then state s_t will move into the next possible state with the transition probability $P(s_t|s_t, a_t)$ while giving a corresponding reward $r(s_t, a_t)$. This process continues over time, and the decision maker can get a sequence of rewards. Clearly, the performance of such systems heavily relies on the quality of hand-crafted features (e.g., state, transition probability).

Application: In view of mobile users, offloading can save their energy and improve computing capability, but it will incur additional transmission and computing resource consumption. Therefore, whether to offload or not should be answered first. If yes, the appropriate MEC server should be selected, and the amount of workload should be determined to offload. In [6], considering the random mobility of a user and the limited capacity of a cloudlet, Y. Zhang *et al.* investigate offloading policy using MDP as an optimization method to achieve a long-term stable offloading/local execution policy. The MDP problem in [6] is solved based on dynamic programming without learning action; however, reinforcement learning algorithms, such as Q-learning, can also be adopted for this problem. The limitation of MDP is that the computational complexity will grow exponentially with the number of states, which leads to the "curse of dimensionality" problem.

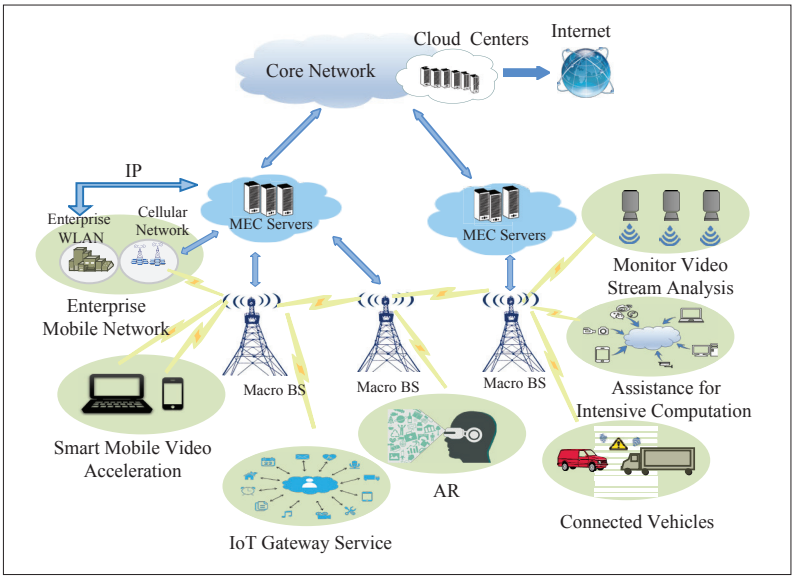

Figure 1. MEC in heterogenous mobile networks and main applications.

Q-Learning: To solve MDP, the system model such as transition probability should be perfectly known. However, in a practical network, this information is very difficult to fully capture, especially in complicated mobile situations. To address the aforementioned problems, Q-learning [5] can be seen as a trial-and-error method to search the optimal policy. The trial and error mean that the decision maker must make a trade-off between exploration and exploitation in an unknown environment. The decision maker prefers to "exploit" the action that has the highest cumulative reward tried in the past, but it also has to "explore" the better new action that may yield the higher reward in the future. Under sufficient learning, the decision maker can finally find the optimal policy, where the goal is to learn the optimal action-value function to obtain the best action for the given state. Compared to the model-based MDP method, Q-learning is a model-free online learning method without any prior environment knowledge.

Application: In a dynamic MEC system, it is infeasible to obtain perfect knowledge of the network environment. Hence, Q-learning may be suitable for decision making with limited information and a dynamic environment, and is usually introduced to design the online and model-free learning approach. To apply Q-learning in MEC, the first step is to identify the actions, states, and reward functions. Then, based on exploration and exploitation, Q-learning can update action-value function by observing the feedback given system states and actions. In [7], L. Xiao *et al.* investigate the mobile offloading problem against smart attacks based on Q-learning in dynamic environments, where some system parameters such as attack cost, offloading gain, and detection accuracy are unknown. Experimental results in [7] illustrate that offloading rate and security using Q-learning are increased by 86 and 6 percent compared to a random offloading scheme,

Since service requirements in MEC are diverse and varied over time, it is beneficial to customize offloading decision for different service features using classification and cluster methods. Besides, these methods can be also used for estimating or predicting radio parameters, handover policy in HetNet, anomaly-detection in wireless network, etc.

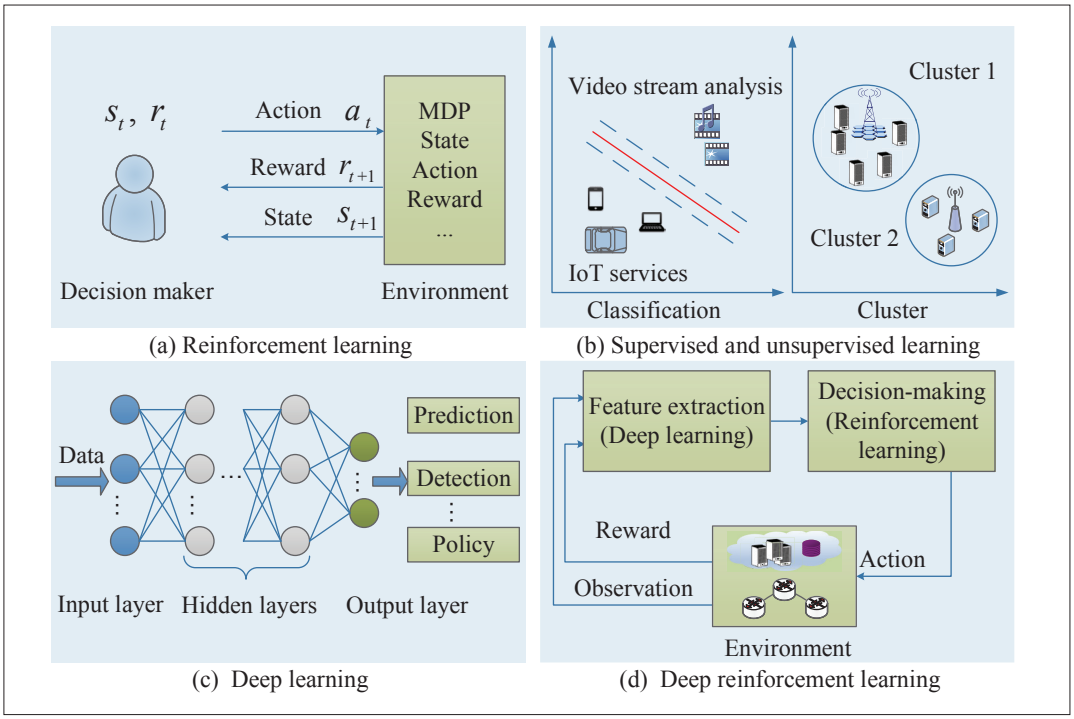

Figure 2. The illustrations of ML approaches: a) reinforcement learning; b) supervised and unsupervised learning; c) deep learning; d) deep reinforcement learning.

respectively. However, the proposed policy cannot guarantee global optimum since all possible states and actions have not been exploited. Besides, similar with MDP, due to "curse of dimensionality," Q-learning is also not applicable to large state space.

Supervised and Unsupervised Learning in Offloading

Supervised learning [8] enables an effective inferring function from labeled training data using statistical rules, which aims to learn an analytical model that can predict the corresponding output results based on input data set. Support vector machine (SVM) and support vector regression (SVR) are the typical supervised learning algorithms, and are widely utilized for discrete-valued classification and continuous-valued regression, respectively. In contrast, the label of input data [9] is unknown in unsupervised learning, and the goal is to find attributes and structures hidden in data to achieve prediction and inferring functions. One of the most widely used unsupervised learning algorithms is *K*-means, which attempts to divide data whose category is unknown into several disjoint clusters. These methods are relatively simple and easy to adopt in practical scenarios, but the performance is sensitive to the training data.

Application: Since service requirements in MEC are diverse and varied over time, it is beneficial to customize offloading decision for different service features using classification and cluster methods. Besides, these methods can also be used for estimating or predicting radio parameters, handover policy in HetNets, anomaly detection in wireless networks, and so on. In [8], A. Khairy *et al.* investigate smart offloading based on supervised learning to optimize execution time and energy consumption for smartphones, where the dataset is built by measuring energy consumption in different contexts. Using SVR, accurate application execution time and energy consumption could be predicted, and a computation offloading decision can be made according to different service features.

Meanwhile, to mitigate interference, a *K*-means-based context-aware mechanism has been adopted in mobile HetNets [9] for small cell clustering. Experimental results in [9] illustrate that the proposed context-aware prediction can guide the mobile user to obtain a stable wireless link. As an inspiration, MEC server selection and offloading decision are made by cluster instead of individual decision, which can be more efficient to decline a number of participants significantly. However, the prediction accuracy is affected by massive data in [8, 9], which makes it hard to theoretically give the performance bound, and thus the performance might be unstable and uncertain. Moreover, a mobile user would be more vulnerable due to classification and clustering, and thus privacy and security should be considered.

Deep Learning in Offloading

Deep learning is a representation (or namely feature) learning method based on a multi-layer neural network, which allows the computational model to automatically extract features needed for prediction or classification from massive raw data. As claimed in [10], the key advantage of deep learning is that these features are learned from data without any manual setup, which is different from reinforcement learning relying on hand-crafted features. However, deep learning is considered as a black box; thus, training tricks and experiences are needed in a practice training model due to no fully theoretical analysis. Since massive data must be trained using complex computing, naturally, introducing MEC could provide available computing resource and raw data.

Application: Deep learning can be seen as an ideal tool for supporting issues including fault and security detection, traffic and behavior prediction, and so on. First, MEC extracts useful features from mobile networks, and the model could be trained on the MEC server side in an offline manner. Then the trained model could provide inferring function for recognition and prediction in a real-time environment. Accordingly, the better understanding of characteristics using deep learning in an MEC system would be helpful to forecast stochastic changes for offloading decision making optimization. C. Liu *et al.* in [11] integrate deep-learning-based classification algorithms into an edge-computing-based real-time computing system. The pre-processed data in an edge device (e.g., a wearable device for healthcare) is transferred to an MEC server. By distributing the computation-intensive task throughout the network, a convolutional neural network (CNN) can be used to perform more accurate data analytics in the proposed system. The proposed recognition method achieves very high detection accuracy (95.2 percent). However, unpredictable training time, long response delay, and massive labeled data are the main challenges on this issue.

Deep Reinforcement Learning in Offloading

As mentioned above, deep learning can extract useful features directly by learning high-dimensional raw data. Meanwhile, traditional reinforcement learning needs hand-crafted features to learn the optimal decision, while low-dimensional state space is required. To learn the optimal control and decision-making in the context of real-world complexity, deep Q-network (DQN) [12] is proposed as typical deep reinforcement learning, which can learn directly from high-dimensional raw data using end-to-end reinforcement learning and deep learning. In DQN, the action-value function is approximated by a deep neural network. In addition, an experience replay mechanism, which can accelerate the training process by randomly sampling stored experiences, is used to update parameters of the deep neural network.

Application: Deep reinforcement learning enables learning a policy in an unknown system through an online approach, which is suitable in a fast changing MEC system. To apply DQN in MEC, the main process follows four steps:

1. First, the operator or controller collects raw data by observation from the edge network, including available computing resource at the MEC server, energy limitation of the mobile user, wireless channel condition, topology, and so on.
2. Second, the filtered data is fed into the DQN model to extract useful features while updating the action-value function.
3. Then the DQN model is trained and updated iteratively by performing an action-value function as feedback.
4. Finally, the DQN model outputs the optimal decision for offloading under certain criteria.

In [13], T. Y. He *et al.* study resource allocation in a connected vehicle network using DQN for base station association, content caching, and offloading polices. By using DQN to approximate and train action-value function, the optimal policy can be obtained with better performance and faster training speed. The performance of the proposed resource allocation policy when networking, caching, and offloading are jointly considered has been improved by 60 percent over those that are not jointly considered. Nevertheless, the larger discrete state space may incur longer training time.

Advantage, Limitation, and Application

In this section, we first conduct a simulation for offloading based on ML and traditional approaches. Second, the main characteristic, advantage, limitation, and application of ML-based approaches are summarized. Finally, we discuss the application of learning methods.

ML Compared to Traditional Approaches

In order to illustrate the advantage and limitation of ML-based approaches, a basic simulation is performed using MDP, Q-learning, and convex optimization for the same offloading problem. Assume that a mobile user with a task moves randomly within the coverage of an MEC server, and the full or partial workload can be transmitted to the MEC server through a wireless network. First, we formulate the offloading problem as an MDP problem to decide how many workloads to offload, and this problem is solved by dynamic programming. Second, as the typical ML-based approach, Q-learning is adopted to determine the optimal policy for the same problem. Third, convex optimization for this offloading problem is also conducted as the benchmark since it is a traditional approach that has been widely used in various scenarios. To evaluate the offloading payoff, it is formulated as the utility function minus the offloading cost and the penalty cost, where the utility function is to show the benefit of offloading (it is a widely used non-decreasing logarithmic function), the offloading cost is caused by the resource consumption in transmission and computation, and the penalty cost is incurred by the offloading failure due to intermittent connection [6] (it is defined as offloading failure probability multiplied by offloaded workload). Let the mathematical expectation of offloading failure probability be 0.1, the corresponding variance be 0.025, the discount factor in MDP and Q-learning be 0.9 (which is to determine the weight of the present and future rewards), and the learning rate be 0.5 for action-value function updating. To show the performance of the learning method, Q-learning is performed based on 1×10^4 and 5×10^4 learning times (denoted as N), respectively.

As shown in Fig. 3a, MDP and Q-learning have higher payoff in the stochastic scenario compared to convex optimization (which generally uses the mathematical expectation of offloading failure probability as a guideline for decision making). Meanwhile, we can also see that the higher number of learning times, the higher payoff of Q-learning. Specifically, we can observe that the performance of Q-learning is unstable when N is small because the state space grows exponentially with the increase of workload. With the more learning times, Q-learning has more opportunities to explore enough states to search the optimal policy. Therefore, Q-learning, as a widely used ML-based approach, is suitable in dynamic and stochastic situations due to its learning ability.

In DQN, the action-value function is approximated by a deep neural network. In addition, the experience replay mechanism, which can accelerate the training process by randomly sampling stored experiences, is used to update parameters of the deep neural network.

Traditional approaches require prior knowledge about mobile user patterns and network parameters. In a static or slowly varying environment, traditional approaches should be preferred. However, in stochastic and fast changing mobile networks, there is no or limited prior knowledge as a guideline for decision making, and thus ML-based approaches should be preferred.

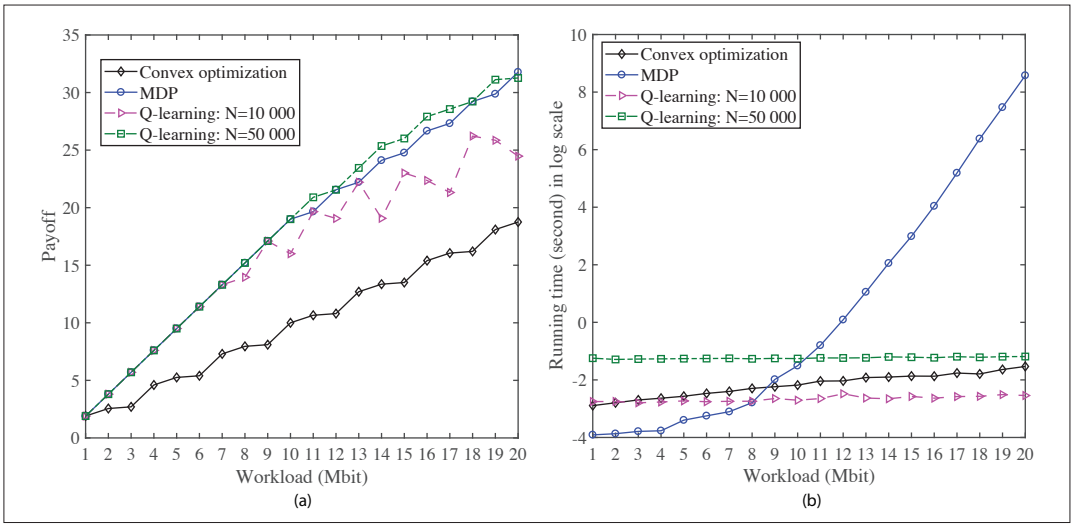

Figure 3. System payoff and running time vs. total workload for offloading using non-learning and learning-based approaches.

Although many ML-based approaches have been proposed for MEC offloading problems, several challenges remain. In particular, the overhead of data gathering, the curse of dimensionality for learning, and computation complexity for decision making are the main obstacles to ML-based approach design. As shown in Fig. 3b, the running time of MDP in log scale grows exponentially with the increase of workload; the reason is that a higher workload will result in larger state space due to discrete states. Moreover, the better performance of Q-learning requires more learning times, which also leads to more computing resource consumption and longer running time.

Table 1 summarizes the basic characteristics, main advantages, limitations, and typical applications of ML-based approaches used in MEC offloading.

ML Compared to Traditional Approaches

It is noted that ML-based and traditional approaches are not conflicting solutions in an MEC system. In contrast, both of them can be used well and complement each other. Traditional approaches require prior knowledge about mobile user patterns and network parameters. In a static or slowly varying environment, traditional approaches should be preferred. However, in stochastic and fast changing mobile networks, there is no or limited prior knowledge as a guideline for decision making, and thus ML-based approaches should be preferred. Moreover, it is computationally infeasible to train over the entire dataset in such an environment. On the other hand, it is necessary to adapt to new patterns dynamically. Therefore, online learning should be designed to obtain an efficient and adaptive policy for MEC over time. To enable quick learning in the case of unknown system parameters, J. Xu *et al.* in [14] propose a novel post-decision state-based reinforcement learning algorithm, which can learn "on the fly" the optimal policy of dynamic workload offloading and edge server provisioning. To address the curse of dimensionality, an online learning algorithm utilizes a decomposition of the offline model iteration and online reinforcement learning, thus improving both the learning rate and the runtime performance significantly.

In summary, the offloading problem under various constraints in MEC needs dynamic online solutions due to the following reasons:

1. The remaining workload at a mobile user varies over time due to offloading, local processing, new arrivals, and timeout departures.
2. The available bandwidth or computing capacity of the target MEC server may change over time due to offloading decisions made by other mobile users.
3. The number of mobile users connected to the target MEC server for offloading may change.
4. The target MEC server may change due to mobile user mobility or mobile network reconfiguration.

AI in MEC System Design: Framework and Challenges

Most existing works using traditional or ML-based approaches in MEC systems only focus on a specific problem or target, and are not generalized and universal for MEC systems, even though some good performance has been shown. To illustrate the general offloading process, a framework for AI in MEC based on the "observe-orient-decide-act" [15] with feedback loop is proposed in Fig. 4. The framework is mainly divided into five tiers from the bottom up, which gradually introduces AI ability according to the service demands and network architecture, named observation, analysis, prediction, policy, and evaluation.

Observation tier: This is the basic function for AI in MEC. It is connected to an MEC system to collect generated or received raw data from the infrastructure layer. In order to quickly extract useful information (e.g., offloading workload, QoS requirement, channel condition, and topology), efficient and effective data acquisition, storage, cleaning, and pre-processing to reduce learning complexity are necessary. Security and privacy are also significant issues that should be addressed in this tier, where

Categories	Reinforcement learning	Supervised/Unsupervised learning	Deep learning	Deep reinforcement learning
Typical techniques	MDP, Q-learning	SVM, SVR/K-means	CNN	DQN
Complexity	• High for large state space	• High for massive data	• High for massive data and multiple layers	• High for large state space
Characteristics	• Learning from own experience • Delayed reward	• Learning from labeled or unlabeled data	• Representation learning from raw data	• Learning control policy directly from high-dimensional data
Advantages	• Learning without a priori knowledge	• Easy and quick to deploy	• End-to-end learning features	• End-to-end reinforcement learning
Limitations	• Curse of dimensionality • Trade-off between exploration and exploitation • Hand-crafted features	• Sensitive to data • Relying on massive data • Hard to theoretically give performance bound	• Very long training time • Training tricks, black-box • Mostly rely on massive labelled data	• Very long training time in large discrete state space • Black-box
Applications	Automatic control and decision • Offloading policy for intermitted cloudlet [6] • Offloading policy against smart attacks [7]	Classification and clustering • Prediction for energy consumption and execution time [8] • Small cell clustering [9]	Detection and prediction • Recognition system for offloading policy [11]	Automatic control and decision • Resource allocation policy for connected vehicles [13]

Table 1. Comparisons of ML-based approaches in MEC offloading.

the unlawful, unsafe, and malicious data in massive raw data should be recognized exactly (e.g., denial of service [DoS] attack). The extracted information is the input of the analysis tier.

Analysis tier: This tier is used for data analysis, in which some useful features can be analyzed based on the filtered data from the observation tier. Due to the proximity to massive data in an MEC system, a real-time inferring function (i.e., classification and cluster functions) should be pre-trained well in this tier to provide guidelines for the next tier. Besides, since some services would require less computational accuracy, so the low-complexity method should be designed to provide quick decision making for MEC with online analysis in a stochastic environment.

Prediction tier: To achieve the exact estimation for offloading considering stochastic and diverse scenarios in MEC, this tier is designed to forecast mobile network changes in the future (e.g., MEC server load, traffic, and user mobility) based on historical experiences and features from the previous tiers. With the aid of this tier, an insightful and foresighted decision could be made in the policy tier to achieve forward-looking planning.

Policy tier: This is crucial for an intelligentized MEC system, which is to make decisions for offloading and other purposes based on analyzed network characteristics and changing trends. Most works in this tier mainly focus on how to formulate models and optimize parameters for specific MEC offloading problems, and thus they are hard to extend in a general MEC system. In fact, the current works only choose an approach to solve a given problem individually depending on experience. Therefore, we believe that the key challenge is to answer "how" to determine which model is appropriate for "what" problems and scenarios, and "when" to use traditional or ML-based approaches.

Evaluation tier: The remote cloud center has the global view of the entire MEC system; thus, it is suitable for training and learning the global strategy in the long term. Meanwhile, with the cooperation between MEC and a remote could center, this tier is to evaluate the effectiveness of policy in terms of energy consumption, resource utilization, QoS, and so on. Accordingly, evaluated results can be fed back to the MEC system to answer whether the previous decision making for offloading is suitable or not, prediction of changing trends is accurate or not, analysis of features and characteristics is exact or not, and extracted data is useful or not. As a result, these tiers can be self-adaptive and self-optimizing to enhance the learning ability of the MEC system iteratively.

Furthermore, each tier in AI function should match those of the MEC system properly. Therefore, future research should carefully investigate

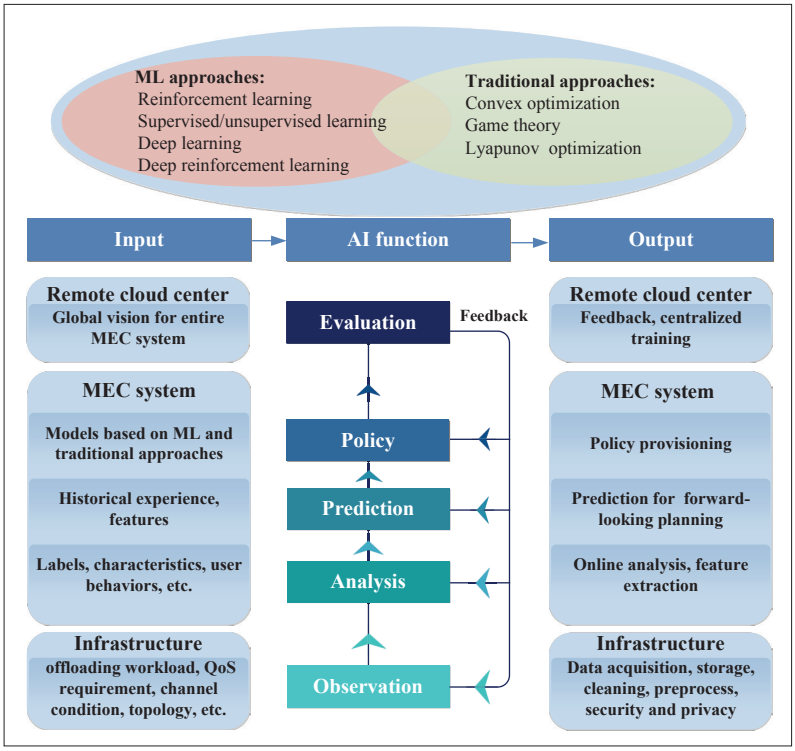

Figure 4. AI in MEC offloading: observe-orient-decide-act with feedback loop.

Each tier cannot optimize its own function and output only, the impact on the next tier should be considered as well. To this end, how to suitably match AI function with the MEC system and jointly optimize among different AI tiers would be an interesting topic in the future.

the relationship and interaction among these tiers. Meanwhile, each tier cannot optimize its own function and output only; the impact on the next tier should be considered as well. To this end, how to suitably match AI function with the MEC system and jointly optimize among different AI tiers would be an interesting topic in the future.

Conclusions

In this article, we have briefly introduced the concept of offloading in MEC and its potential benefits. We have illustrated the necessity for AI in MEC and discussed basic ideas of typical ML-based approaches in the state-of-the-art research work. Next, we have compared the performance of ML-based approaches with that of traditional approaches. Meanwhile, the main characteristics, limitations, and applications have been illustrated. Even though existing research has shown great benefits of AI in MEC, some potential research topics still remain. Therefore, we show our preliminary idea about how to design an AI-based MEC system, and discuss some challenges for intelligent offloading in MEC.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants 61701059 and 61671096, and in part by the Research and Innovation Project of Graduated Students of Chongqing under Grant CYS17218.

References

[1] T. X. Tran *et al.*, "Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, Apr. 2017, pp. 54–61.
[2] ETSI, "MEC in 5G Networks," white paper, no. 28, June 2018.
[3] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Commun. Surveys & Tutorials*, vol. 19, no. 3, Mar. 2017, pp. 1628–56.
[4] C. Jiang *et al.*, "Machine Learning Paradigms for Next-Generation Wireless Networks," *IEEE Wireless Commun.*, vol. 24, no. 2, Apr. 2017, pp. 98–105.
[5] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," *IEEE Trans. Neural Net.*, vol. 9, no. 5, Sept. 1998, pp. 1054–54.
[6] Y. Zhang, D. Niyato, and P. Wang, "Offloading in Mobile Cloudlet Systems with Intermittent Connectivity," *IEEE Trans. Mobile Comp.*, vol. 14, no. 12, Dec. 2015, pp. 2516–29.
[7] L. Xiao *et al.*, "Mobile Offloading Game Against Smart Attacks," *Proc. IEEE INFOCOM Wksps.*, 2016, pp. 403–08.
[8] A. Khairy, H. H. Ammar, and R. Bahgat, "Smartphone Energizer: Extending Smartphone's Battery Llife with Smart Offloading," *Proc. IEEE IWCMC*, 2013, pp. 329–36.
[9] M. Xiong and J. Cao, "A Clustering-Based Context-Aware Mechanism for IEEE 802.21 Media Independent Handover," *Proc. IEEE WCNC*, 2013, pp. 1569–74.
[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, May 2015, pp. 436–44.
[11] C. Liu *et al.*, "A New Deep Learning-Based Food Recognition System for Dietary Assessment on An Edge Computing Service Infrastructure," *IEEE Trans. Serv. Comp.*, vol. 11. no. 2, Mar. 2018, pp. 249–61.
[12] V. Mnih *et al.*, "Human-Level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, 2015, pp. 529–33.
[13] Y. He, N. Zhao, and H. Yin, "Integrated Networking, Caching and Computing for Connected Vehicles: A Deep Reinforcement Learning Approach," *IEEE Trans. Vehic. Tech.*, vol. 67, no. 1, Jan. 2018, pp. 44–55.
[14] J. Xu, L. Chen, and S. Ren, "Online Learning for Offloading and Autoscaling in Energy Harvesting Mobile Edge Computing," *IEEE Trans. Cognitive Commun.*, vol. 3, no. 3, Sept. 2017, pp. 361–73.
[15] ETSI, "Improved Operator Experience Through Experiential Networked Intelligence (ENI)," white paper, no. 22, Oct. 2017.

Biographies

Bin Cao (caobin65@163.com) received his Ph.D. degree in communication and information systems from the University of Electronic Science and Technology of China, Chengdu, in 2014. Currently, he is an associate professor with the College of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, China. His research interests include cooperative communications, wireless network virtualization, and mobile cloud computing.

Long Zhang (zhanglong3211@yeah.net) is pursuing his Master's degree at the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications. His research interests include mobile edge computing and the Internet of Things.

Yun Li (liyun@cqupt.edu.cn) received his Ph.D. degree in communication engineering from the University of Electronic Science and Technology of China. Currently, he is a professor of electrical engineering with the College of Communication and Information Engineering, Chongqing University of Posts and Telecommunications. His research interests include mobile cloud computing, cooperative/relay communications, and virtual wireless networks.

Daquan Feng (fdquan@gmail.com) received his Ph.D. degree in information engineering from the University of Electronic Science and Technology of China in 2015. Currently, he is an assistant professor with the College of Information Engineering, Shenzhen University, China. His research interests include D2D communications, LTE-U, and massive IoT networks.

Wei Cao (weicao@princeton.edu) is pursuing her Ph.D. degree at the University of Electronic Science and Technology of China and is currently a visiting student research collaborator in the Department of Electrical Engineering, Princeton University. Her research interests include theoretical analysis, optimization, and algorithmics in the next generation cellular network for massive machine-type communication and the Internet of Things.