Edge Computing with Artificial Intelligence: A Machine Learning Perspective

HAOCHEN HUA, Hohai University, P. R. China
YUTONG LI, Tsinghua University, P. R. China
TONGHE WANG, Guangzhou Institute of Energy Conversion, P. R. China
NANQING DONG, University of Oxford, United Kingdom
WEI LI, University of Sydney, Australia
JUNWEI CAO, Tsinghua University, P. R. China

Recent years have witnessed the widespread popularity of Internet of things (IoT). By providing sufficient data for model training and inference, IoT has promoted the development of artificial intelligence (AI) to a great extent. Under this background and trend, the traditional cloud computing model may nevertheless encounter many problems in independently tackling the massive data generated by IoT and meeting corresponding practical needs. In response, a new computing model called edge computing (EC) has drawn extensive attention from both industry and academia. With the continuous deepening of the research on EC, however, scholars have found that traditional (non-AI) methods have their limitations in enhancing the performance of EC. Seeing the successful application of AI in various fields, EC researchers start to set their sights on AI, especially from a perspective of machine learning, a branch of AI that has gained increased popularity in the past decades. In this article, we first explain the formal definition of EC and the reasons why EC has become a favorable computing model. Then, we discuss the problems of interest in EC. We summarize the traditional solutions and hightlight their limitations. By explaining the research results of using AI to optimize EC and applying AI to other fields under the EC architecture, this article can serve as a guide to explore new research ideas in these two aspects while enjoying the mutually beneficial relationship between AI and EC.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computing methodologies** → **Artificial intelligence**; • **Computer systems organization** → **Distributed architectures;**

Additional Key Words and Phrases: Edge computing, artificial intelligence, machine learning

This work is supported in part by the National Natural Science Foundation of China under Grant No. 52107089, in part by China Postdoctoral Science Foundation under Grant No. 2021M70040, in part by the Fundamental Research Funds for the Central Universities of China under Grant No. B200201071, and in part by the BNRist Program under Grant No. BNR2021TD01009.
Authors' addresses: H. Hua, College of Energy and Electrical Engineering, Hohai University, Nanjing, Jiangsu, P. R. China, 211100; email: huahc16@tsinghua.org.cn; Y. Li, Department of Automation, Tsinghua University, Beijing, P. R. China, 100084; email: liyt19@mails.tsinghua.edu.cn; T. Wang, Micro Energy Research Group, Guangzhou Institute of Energy Conversion, Chinese Academy of Sciences, Guangzhou, Guangdong, P. R. China, 510640; email: wangth@ms.giec.ac.cn; N. Dong, Department of Computer Science, University of Oxford, Oxford, England, United Kingdom, OX1 3QD; email: nanqing.dong@cs.ox.ac.uk; W. Li, Centre for Distributed and High Performance Computing, School of Computer Science, University of Sydney, Sydney, New SouthWales, Australia; email: weiwilson.li@sydney.edu.au; J. Cao (corresponding author), Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, P. R. China, 100084; email: jcao@tsinghua.edu.cn. Q1

0360-0300/2022/08-ART00 $15.00
https://doi.org/10.1145/3555802

ACM Reference format:
Haochen Hua, Yutong Li, Tonghe Wang, Nanqing Dong, Wei Li, and Junwei Cao. 2022. Edge Computing with Artificial Intelligence: A Machine Learning Perspective. *ACM Comput. Surv.* 00, JA, Article 00 (August 2022), 35 pages.
https://doi.org/10.1145/3555802

---

1 INTRODUCTION

Cloud computing has been widely used since its inception and has greatly changed people's lifestyle. Many large companies, including Google, Amazon, and Microsoft, have launched their own cloud computing services (Google Cloud, Amazon Web Services, Microsoft Azure, respectively). Equipped with a large number of remotely located servers, cloud computing can intelligently provide users with computing, storage, and network services in real time according to user needs in terms of resource type, quantity, and so on [1]. In this case, users can easily obtain these cloud services with a small fee or totally for free [2].

1.1 Edge Computing

The development of **Internet of things (IoT)** has driven the production and application of a large number of hardware devices/sensors worldwide. These hardware devices/sensors have the ability to sense the surrounding physical environment and transform the environmental information into data. After these massive data are transmitted to the cloud for computing or storage, data consumers can access cloud data according to their individual needs and then extract the information they need [3].

However, with the continuous development and widespread application of IoT, cloud computing has begun to expose more and more problems. For instance, if the data generated by global terminal devices are computed and stored in a centralized cloud, then it will cause a series of problems, including low throughput, high latency, bandwidth bottlenecks, data privacy, centralized vulnerabilities, and additional costs (such as transmission cost, energy cost, storage cost, calculation cost). In fact, many application scenarios in IoT, especially **Internet of vehicles (IoV)**, have requirements of high speed and low latency for data processing, analyzing, and result returning [4].

To address these challenges of cloud computing mentioned above, a new computing paradigm, called **edge computing (EC)**, has attracted widespread attention. Simply put, the core idea of the EC model is to offload the data processing, storage, and computing operations that were originally required by the cloud to the edge of the network near terminal devices. This helps to reduce data transmission time and device response times, reduce the pressure on network bandwidth, reduce the cost of data transmission, and also achieve decentralization [5].

1.2 Artificial Intelligence

Artificial intelligence (AI) is a kind of technology that endows the machine with certain intelligence so that the machine has the same ability to solve tasks as human beings [6]. While heuristic-based algorithms and **data mining (DM)** [7] have both played an important role in AI solutions to IoT in the past decades, we mainly focus on **machine learning (ML)**, a recently popular area in AI. It is worth mentioning that, though DM and ML share similarities in utilizing massive data, ML focuses on mimicking the human learning process, but DM is designed to extract the rules from data [8, 9]. In contrast to DM, ML is a higher-level intelligence and represents the future direction of AI.

The widespread application of AI, especially ML, has clearly become an inevitable trend in the "big data era" brought by IoT. It is worth noting that this article focuses on the new generation AI algorithm, e.g., **deep learning (DL)**, and so on. Note that some of these applications have high requirements for latency and network stability, but these requirements are often not guaranteed by cloud computing. In contrast, the new EC model can meet these requirements by deploying AI at the edge and delegating some computing and storage resources to edge devices close to the terminal. Although EC brings benefits such as reduced latency, improved data privacy, and enhanced security, the limited computing and storage capacity of edge devices has brought new problems. Using AI to optimize EC and solve the problems faced by EC has become a new trend in related research [10].

1.3 Combination of Edge Computing and Artificial Intelligence

The motivations of combining AI and EC in recent works can be roughly divided into two aspects, which fully illustrate the mutual benefit between AI and EC:

(1) The development of EC still faces many challenges, e.g., task scheduling, resource allocation, delay optimization, energy consumption optimization, and privacy and security. In response, many researchers have adopted AI-based solutions to promote the development of EC.
(2) In spite of the rapid development of AI, its application relies on strong computing power. Traditional cloud computing can provide abundant computing and storage resources, but cloud-based AI reasoning and training may lead to significant delay as well as data privacy and security issues. By executing AI tasks in edge nodes closer to the user side, EC can greatly alleviate the aforementioned issues with improved stability, reliability, and user experience.

At present, researchers have made many great achievements in the above research problems. This article summarizes these results, hoping that readers can quickly get updated with the latest research status and relevant results.

1.4 Review of Existing Surveys

EC and AI are very popular research fields, and some related reviews have been published. In Reference [11], authors focus on the motivation and research work of deploying AI algorithm on the edge of the network. The latest development of ML in mobile EC is reviewed in Reference [12], which includes the development of 5G network in automatic adaptive resource allocation, mobility modeling, security, and energy efficiency. Survey work [13] reviews the application of DL in EC, and it focuses on how to use DL to promote the development of edge applications, e.g., intelligent multimedia, intelligent transportation, intelligent city, and intelligent industry. Various methods of fast implementation of DL reasoning in the combination of end devices, edge servers and cloud, and the methods of training DL models in multiple edge devices are also discussed in Reference [14]. To achieve the best performance of DL training and reasoning, Reference [15] comprehensively discusses how to design EC architecture with communication, computing power, and energy consumption constraints. From the perspective of algorithms and systems, [16] csystematically summarizes the latest approaches to overcome the communication challenges caused by AI reasoning and training at the edge of the network.

Nonetheless, the mutually beneficial relationship between EC and AI (especially traditional ML, DL, **reinforcement learning (RL)**, and **deep reinforcement learning (DRL)**) are seldom discussed in previous surveys. From this point of view, this article reviews existing works on EC performance optimization and different application scenarios of AI. In addition to the DL methods discussed in References [13–15], other ML algorithms, especially RL and DRL, are also discussed in this article.

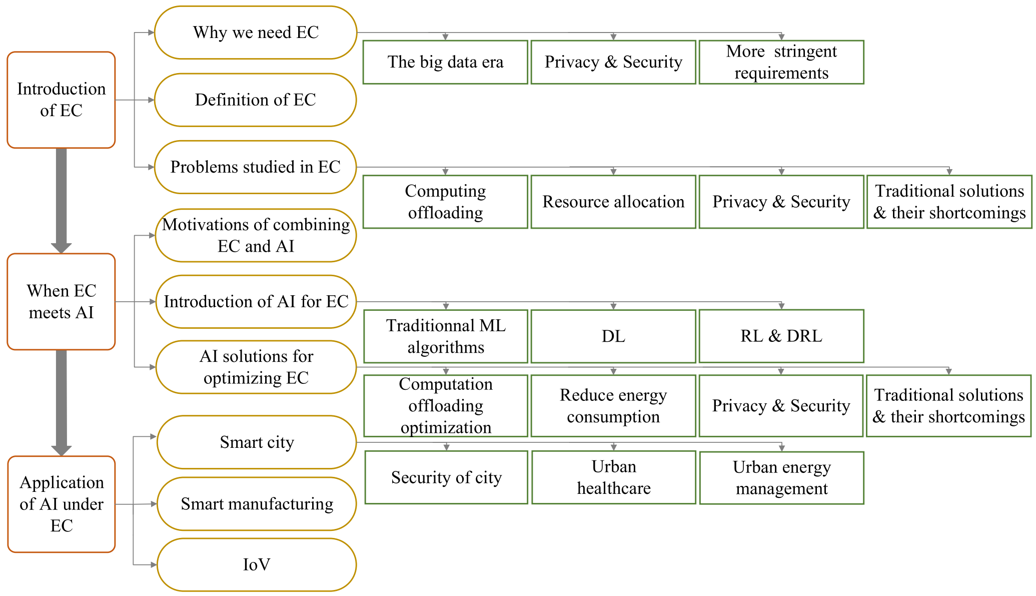

Fig. 1. Structure of the survey.

1.5 Our Contributions

Our main contributions in this article are as follows:

(1) We first outline the basic definition and architecture of EC and discuss the necessity of EC in the presence of cloud computing. We also describe the problems studied by EC.
(2) We discuss the motivations for combining AI and EC from two perspectives:
   - AI algorithms can be utilized to optimize EC;
   - EC enables AI to be deployed on the edge to bring faster response speeds and network stability for AI applications in different fields.

   We summarize three ideas of deploying AI training and reasoning tasks in the EC architecture based on existing studies and analyze their advantages and disadvantages.
(3) We mainly introduce popular ML algorithms in the field of AI and analyzes their respective advantages. We summarize the latest research on solving the problems of EC and optimizing the performance of EC by using AI algorithms. We also review the latest research on applying AI to other fields under the EC architecture.

Roadmap. The remainder of this article is organized as follows: Section 2 introduces the definition of EC, discusses why we need EC, and enumerates the challenges faced by EC and corresponding traditional (non-AI) solutions. In Section 3, we combine EC and AI. We first discuss the trends and reasons for the combination of the two, then introduce the corresponding AI algorithms, and finally conduct a comprehensive review of the research on using AI algorithms to optimize EC. In Section 4, we summarize recent works on applying AI to other fields under EC. We summarize this article in Section 5. The diagram in Figure 1 shows a clear picture of the structure of this article.

2 INTRODUCTION OF EDGE COMPUTING

Cloud computing has been a very popular or even a household concept for the past decade. Cloud computing brings many conveniences. For example, small- and medium-sized enterprises only need to purchase cloud server resources at a relatively low cost, without the need of purchasing their own hardware and equipment at high prices. This greatly reduces the cost of business operations and the threshold for companies to engage in technology research and development.

The centralized computing, storage, and network resources of cloud computing has exposed a series of problems with the development of the times. In this context, EC, a new computing paradigm, has begun to attract the attention of all areas. In this section, we will give a brief overview of EC. We will first discuss why EC is needed, and then introduce what EC is. Finally, we will discuss the problems of EC and corresponding traditional solutions, and point out the shortcomings of these traditional solutions.

2.1 Why We Need Edge Computing

We will explain the necessity of EC from the following three aspects: the "big data era" caused by IoT, more stringent requirements of high network stability and response speed, and the consideration of privacy and security.

2.1.1 The Big Data Era Caused by Internet of Things. The concept of IoT was proposed in 1999 for supply chain management, but now IoT covers a much wider area [17]. With the integration of IoT into traditional industries, many new application areas have been spawned, such as smart home, smart grid, smart traffic, and intelligent manufacturing. The idea of IoT is that things connected to the Internet form a huge network, achieving the interconnection of these things at any time and place. With the continuous development of IoT, the number of various sensors, smartphones, healthcare applications and online social platforms is soaring, and the resulting global data will increase to 175 **zeta bytes (ZB)** by 2025 according to the prediction of **International Data Corporation (IDC)** [18]. This huge data volume has facilitated the world of big data [19].

In the era of big data, the most direct and simple method for handling those data is to transfer the data to the cloud for processing. The annual global cloud IP traffic of 2016 was 6.0 ZB, and it is expected to reach 19.5 ZB in 2021, reported by Cisco in 2018 [20] . However, the computing power of the cloud is increasing linearly [21], which is much slower than the current rate of data growth. With the rapid growth of data, cloud computing will no longer be fully trusted.

2.1.2 More Stringent Requirements of Network Stability and Response Speed. There are some IoT application scenarios that require extremely fast response speeds. For example, in the scenario of intelligent driving, sensor devices such as cameras are installed in autonomous vehicles. These sensor devices can continuously obtain data from the surrounding environment during the autonomous driving mode. In the cloud computing model, these data will be uploaded to the cloud for computing, and the results will be returned back to the vehicle's control chip. Considering the complicated driving environment of a vehicle, this method is actually very time-consuming, and it may even cause the smart vehicle to fail to make the right decision in a timely manner, resulting in serious consequences [3].

In the fields of **augmented reality (AR)** and **virtual reality (VR)**, mobile AR/VR applications need to continuously transmit high-resolution videos, so they have high requirements for data computing capabilities, network stability, and response speed [22]. At the current rate of data growth, the cloud's computing power becomes less and less proficient in meeting these requirements. However, uploading all the data to the cloud will cause serious network congestion. Due to the limited network bandwidth, the data generated by a large number of IoT devices will impose a lot of pressure on the network bandwidth, causing cloud computing to no longer meet the requirements of latency and response speed in these scenarios. In addition, these data may have a large proportion of noise and errors. Some survey shows that only one third of the data obtained by most sensors are correct [23]. Putting these worthless data into the cloud will cause a huge waste of cloud server resources and a waste of network bandwidth.

2.1.3 Privacy and Security. Cloud computing has outsourcing features. Users need to host local data to the cloud when using cloud computing. This leads to a series of data security and privacy issues [21]. The data loss during long-distance transmission between devices and the cloud can damage the integrity and accuracy of the data. In addition, highly centralized computing and storage can also become serious problems. When one device in a centralized system goes wrong due to benign errors or malicious attacks, other devices will be negatively affected. The data privacy problem refers to the theft and utilization by other unauthorized persons, companies or organizations. Actually, data owners have lost control of their data uploaded to the cloud, so it is difficult to guarantee data privacy [24].

2.2 The Definition of Edge Computing

The origin of EC can be traced back to 1999 when Akamai proposed **content delivery networks (CDN)** for web page caching near the clients, aiming to improve the efficiency of web page loading [25]. The concept of EC was borrowed from the cloud computing infrastructure to expand the concept of CDN [26].

EC now has many different definitions. For example, Openstack defines EC as a model that provides application developers and service providers with cloud services and IT environmental services at the edge of the network [27]. In Reference [28], the authors believe that the “edge” in EC refers to any computing and network resources between the data source and the cloud, such as smart phones, gateways, micro data center, and cloudnet. It can also be understood that EC offloads some cloud resources and tasks to the edge near users and data sources.

It should be noted that EC cannot replace the roles and advantages of cloud computing due to the indispensable computing power and storage capacity of the cloud. The emergence of EC is to make up for the limitations of cloud computing, and the relationship between EC and cloud computing should be complementary. Therefore, how to coordinate the relationship between the cloud and the edge so that the two can cooperate more efficiently and securely is a problem that needs to be studied.

EC’s general architecture is three-layered, as shown in Figure 2, which are end, edge, and cloud [29].

- *End.* This layer has two main functions. The first is to perceive the world, which is to observe, obtain and digitize the information of the physical world. This function is completed by various types of sensors, such as speed sensors on smart cars, or cameras in smart cities. The second is to receive information or data from the edge or cloud and perform the corresponding tasks. Data obtained from the end is processed by the edge and the cloud, and then the results will be fed back to the end according to user needs, such as control signals in smart driving or video traffic accepted by smartphones. Devices in this layer may have some but very limited computing and storage capabilities.
- *Edge.* The edge layer is between the cloud and the end. This layer contains certain computing, storage, and network resources, so some tasks that were originally performed in the cloud can be delegated to this layer for execution. Since this layer is closer to end devices, EC has the advantages of low latency. Generally, the edge layer is composed of gateways, control units, storage units, and computing units.
- *Cloud.* This layer actually refers to cloud servers that has been widely used in practice. In addition to its powerful computing and storage capabilities, the cloud also has the ability to macro-control the entire EC architecture.

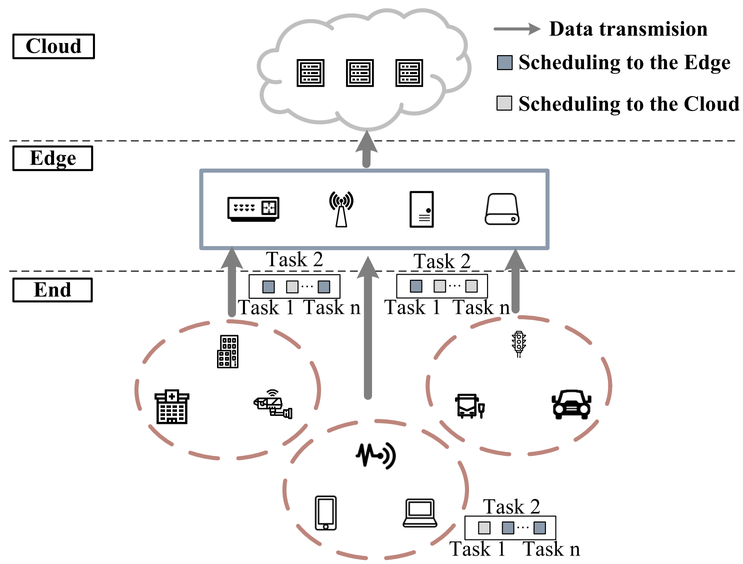

Fig. 2. Architecture of EC. Gray arrows indicate the data transmission between the end, the edge, and the cloud. Blue and gray boxes indicate that the task is scheduled to the edge and the cloud, respectively.

EC has advantages in offloading some resources and tasks on the cloud to the edge. The edge layer is closer to end users and data source, so the transmission distance is greatly shortened, and the corresponding transmission time is greatly reduced. This effectively improves the response speed of user requests. At the same time, the shortened transmission distance also reduces the cost and data security issues caused by the long-distance transmission. From the perspective of the cloud, large-scale raw data will be processed on the edge to filter out a large number of useless and erroneous data first, and then the edge uploads important data or information to the cloud. This greatly reduces the bandwidth pressure, the transmission cost, and the possibility of user privacy leakage.

2.3 Problems Studied in Edge Computing

Next, we will describe three problems studied in the field of EC in detail: computing offloading, resource allocation, and privacy and security. We will also explain the shortcomings of traditional solutions to these problems.

2.3.1 Computing Offloading. Computation offloading was originally proposed in cloud computing. The definition is that the terminal devices with limited computing power delegates part or all of the computing tasks to the cloud for execution. Similarly, computing offloading in EC refers to the problem that terminal devices with limited computing power delegate part or all of its computing tasks to the edge [30]. The main considerations are whether terminal devices will offload, how much they will offload and to which nodes they will offload. Computing offloading solves the problems of insufficient resources and high energy consumption in terminal devices.

Traditional methods of computing offloading applied to cloud computing are based on many assumptions, including that the default server has sufficient computing power and does not care about its energy consumption or network condition. However, traditional methods based on the above assumptions are not suitable for solving the computing offloading in EC where edge devices and servers have limited computing capabilities [31]. Reasonable computing offloading strategies are able to reduce energy consumption and latency. Therefore, computing offloading is an important research topic for optimizing EC.

2.3.2 Resource Allocation. Compared to traditional cloud computing, the most prominent advantage of EC is that it does not need to upload all the data to the cloud for computing and storage tasks, which largely frees up network bandwidth and other resources occupied by cloud computing. In the meanwhile, since tasks are distributed on each edge node with limited resources, an intelligent and efficient solution for resource management is crucial for EC.

2.3.3 Privacy and Security. EC also faces new challenges regarding data security and privacy [32]. Some of these challenges come from the inherent problems of cloud computing, and others come from the distributed and heterogeneity nature of EC itself [33]. Traditional solutions for data security and privacy issues of cloud computing are not applicable to the non-centralized computing model of EC. Therefore, further improving data security and further protecting data privacy is a problem worthy of researchers' attention.

2.4 Summary

Aiming at the problems described above, many studies based on traditional methods have made good progress. In solving the problem of resource allocation and computing offloading in EC, some researchers adopt Lyapunov optimization algorithm [34] to find the optimal decision [35, 36]. Some studies also regard resource allocation and computing offloading as optimization problems such as linear programming [37] and mixed integer non-linear programming [38–40]. Other traditional methods include **alternating direction method of multipliers (ADMM)** [41], Stackelberg game [42], and so on. In terms of security, Jing et al. [43] adopt a linear programming method to reduce data loss. Kang et al. [44] use blockchain technology to protect the security of data storage and sharing. In terms of privacy protection, traditional methods include differential privacy [45], wavelet transform [46], and so on.

Although traditional methods above have achieved good results in optimizing EC, they still have some shortcomings. First, the underlying model needs to be known, which is not an easy task due to the complexity and dynamics of EC itself. Second, they are easy to converge to local optima, and their efficiency is usually very low. Moreover, they lack the ability to perform deep and high-dimensional data mining, automatically extract important features to make fast optimal decisions, and make prediction. Note that these are all advantages of AI algorithms, and we will describe how they optimize EC in the next section.

In summary, this section mainly focuses on the concept and motivation of EC. At the same time, the problems and challenges faced by the development of EC are also described. It is worth noting that traditional methods have achieved good results in solving these problems, but they still suffer some shortcomings. In the future, AI algorithms might become more adaptable to new situations, able to change inputs, outputs, and constraints more easily, and do not need mathematical models when data are sufficient [12].

3 WHEN EDGE COMPUTING MEETS ARTIFICIAL INTELLIGENCE

In this section, we will first analyze the respective development of AI and EC and the motivation for the combination of the two, and then we will give an overview of related AI algorithms. Finally, we will summarize AI-based algorithms for topics such as computing offloading optimization, non-computing offloading methods to reduce energy consumption, EC security, data privacy, and resource allocation optimization.

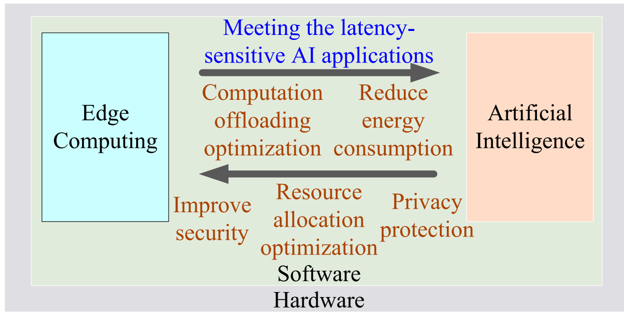

Fig. 3. Mutually beneficial relationship between AI and EC. The right-to-left arrow indicates that the optimization and development of EC require the assistance of AI algorithms (e.g., computation offloading optimization). The left-to-right arrow indicates that EC needs to be deployed closer to terminal devices to meet the requirements of some latency-sensitive AI applications (e.g., smart city).

3.1 Motivations of Combining Edge Computing and Artificial Intelligence

Artificial intelligence is a very critical technology in the era of big data. It brings intelligence and reasoning capabilities to a large number of terminal devices in IoT. At present, many studies and applications have combined the two hot areas of AI and EC, and their motivations can be roughly divided into two aspects:

- The optimization and deployment of EC requires the assistance of AI algorithms;
- EC provides necessary computing functions for AI applications that need to be deployed close to terminal devices for low latency and high network stability [47].

It can be seen that the development of AI and EC is mutually beneficial (see Figure 3 for a straightforward description), and the combined development of the two has attracted the attention of many researchers.

3.1.1 Edge Computing Benefits Artificial Intelligence. In detail, EC brings benefits to the application of AI. With the advent of the big data era, the widespread application of AI in people's daily lives has become an irresistible trend. Of course, this trend still faces challenges. For example, AI's reasoning and training requires strong computing power and sufficient energy support, but terminal devices often do not meet these two requirements. In recent years, cloud computing has fulfilled these needs by offloading AI model training and reasoning tasks that terminal devices cannot perform to the cloud server. However, relying solely on cloud computing will cause problems like insufficient bandwidth and high latency when a large number of AI models are used by a large number of terminal devices [48]. With the advent of EC, AI can be deployed near terminal devices and users on the edge and terminal with certain computing resources and storage resources, therefore meeting the needs for low latency and high network stability [11].

In return, EC also brings three ideas to the application of AI in other fields (visually represented by Figure 4).

(a) Massive data are preprocessed and then uploaded to the cloud for AI training and reasoning [49]. Although this idea has greatly reduced the pressure of massive data on bandwidth and transmission costs, it does not meet the requirements of many applications in terms of latency (e.g., IoV and AR/VR applications).
(b) To reduce the latency of applications, AI reasoning tasks are performed on the edge or the end, while model training tasks are still performed in the cloud [50].

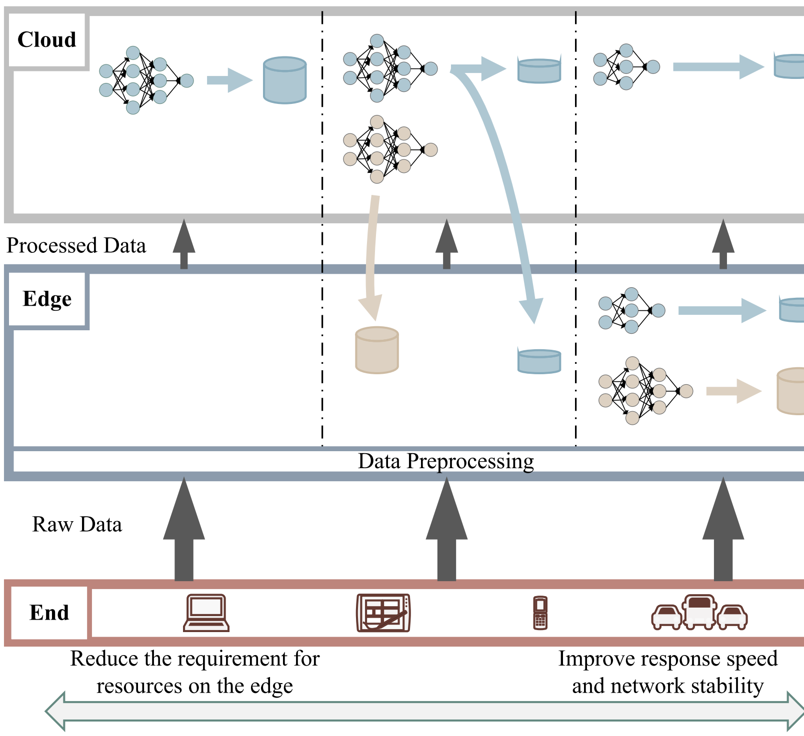

Fig. 4. Hierarchical modes for deploying AI in EC. The figure is divided into three parts by two vertical dotted lines, which correspond to three hierarchical modes. Neural networks and cylinders represent training tasks and reasoning tasks, respectively. (a) The leftmost part describes that both training and reasoning tasks are deployed in the cloud. (b) The blue part in the middle describes that the training tasks are performed in the cloud, but the reasoning tasks are performed in both cloud and edge. The red part in the middle describes that the training tasks are in the cloud, while the reasoning tasks are performed completely on the edge. (c) The blue part in the rightmost part indicates that both training and reasoning tasks are deployed in both cloud and edge. The red part describes the training and reasoning tasks performed only on the edge.

(c) Delegate part or all of AI training and reasoning tasks to the edge [51]. With distributed characteristics, this idea helps enhance the location awareness of AI models while reducing the latency and bandwidth pressure [33]. Note that the requirements for energy consumption and computing power of edge devices will also increase as the number of tasks devolved to the edge side increases.

As can be seen from the above, these three ideas have their own advantages and disadvantages, so existing studies are more inclined to choose the best idea according to the specific situation.

3.1.2 Artificial Intelligence Benefits Edge Computing. AI is playing an important role in the optimization of EC [52]. Since EC is distributed and the workload of each edge device changes dynamically with time and location, this uncertainty and unpredictability have brought huge obstacles to the application of EC. In this sense, EC still needs to be optimized and improved in many aspects, such as optimizing computing offloading, optimizing resource allocation, reducing latency and energy consumption, and improving user experience.

Many optimization problems in EC are very complex non-convex problems. As the number of devices and users increases, the scale of these problems will also rapidly increase [53]. Compared to traditional methods, ML is more suitable for solving optimization problems of EC and has better results [54]. In addition, AI algorithms are also good at effectively mining hidden information and laws from data in complex and noisy EC environments, which has plagued traditional optimization methods for a long time.

3.2 Introduction of Artificial Intelligence Algorithms in Edge Computing

We are going to introduce these AI algorithms used in EC, namely, traditional ML algorithms, DL, RL and DRL algorithms. We will also provide some examples of application accordingly. In this article, we mainly focus on the field of ML in AI algorithm. Other algorithms such as evolutionary algorithm are not the focus of this article, but are briefly introduced in this section.

3.2.1 Traditional Machine Learning. The traditional ML algorithms in this work particularly refer to those ML algorithms other than DL and RL. Given the availability of label information, the traditional ML algorithms can be divided into supervised learning, semi-supervised learning, and unsupervised learning. Among them, supervised learning requires labeled data to train the model, while unsupervised learning can autonomously discover the principles implicit in the data. As a hybrid of supervised learning and unsupervised learning, semi-supervised learning has access to both labeled data and unlabeled data. For example, the common supervised learning methods include **support vector machines (SVM)**, boosting, and random forests; the common semi-supervised learning methods include label propagation and graphical models; the common unsupervised learning methods include clustering algorithms such as K-means and dimension reduction algorithms such as **principal component analysis (PCA)**.

There are some obvious shortcomings of traditional ML algorithms. For instance, they are sensitive to data sets, the data become less effective when the data set is large enough, and they need complicated artificial feature engineering. In spite of these shortcomings, traditional ML has small energy consumption, small computing power cost, and is easy to deploy compared to DL and RL. Due to the distributed nature of EC, the appropriate AI algorithm can be reasonably selected according to the resource situation and task requirements of each edge and terminal device, so traditional ML can also rely on these advantages to find its place in EC [55].

3.2.2 Deep Learning. DL resembles the functions of human brains. It has the ability to autonomously learn high-level features from raw data, thereby efficiently performing classification and prediction tasks [56, 57]. DL is usually deployed in a multi-layer structure. These layers can be fully connected layers, convolutional layers, pooling layers, normalization layers, or activation layers. A DL algorithm can be formed by the free combination of these layers. The more layers the algorithm includes, the “deeper” it is. The input of a neuron in each layer is the weighted sum of the outputs of the neurons in the previous layer. After the input is activated by an activation function, the obtained number is used as the output of the neuron [58]. Compared to traditional ML algorithms, DL has a more powerful ability to extract high-level features from massive data due to its multilayer structure [59].

The common DL models include: **deep neural networks (DNN)**, **convolutional neural networks (CNN)**, **recurrent neural networks (RNN)**, and so on.

- DNN, also known as **multiple linear perceptrons (MLP)**, is a neural network with multiple hidden layers. The neural network layer in DNN can be divided into three types: input layer, hidden layer and output layer. By adding hidden layers, DNN model can obtain more powerful learning ability.
- CNN is composed of a series of different convolution layers. High-level features hidden in the input data can be extracted through the convolution operation in these convolution

layers [60]. CNN has powerful representation abilities and picture recognition capabilities. Based on this, some studies have adopted CNN algorithms in the fields of fault detection and video surveillance in EC. For example, Zhang et al. [61] detects microseismic events by deploying CNN models on edge devices.

- RNN is a DNN algorithm that is good at modeling and processing sequence data. However, a major disadvantage of RNN is that it is easy to forget. That is, the impact of the input of the starting moment on the later moments will become smaller and smaller with time. Therefore, an improved version of RNN named **long short-term memory (LSTM)** [62] is proposed. At present, some studies [63–65] have adopted the LSTM algorithm to solve the issues faced by EC.

When a large number of labeled data are available, compared with traditional ML algorithms, DL performs better in natural language processing, computer vision and many other fields [57]. The characteristics of EC make the data collected from the physical environment can be processed locally, which meets the requirements of DL. Therefore, some EC studies also focus on using DL in EC anomaly detection [66], task scheduling and resource allocation in EC [67], and privacy protection [68].

3.2.3 Reinforcement Learning and Deep Reinforcement Learning. Unlike supervised learning and unsupervised learning that rely on static data, RL is a learning algorithm that trains models through dynamic interaction with the environment. The core idea is that agents receive the state of environment and make actions to maximize the reward according to historical experience. Because reinforcement learning is good at solving decision-making problems, some studies [69, 70] have adopted RL algorithm in the decision-making of EC resource management, allocation, and scheduling.

Typical algorithms in RL are model-free and value-based Q-learning algorithm [71]. Each iteration of Q-learning algorithm will calculate an expected cumulative reward, called the Q-value, according to current state and given action. However, as the environment becomes more complex, the state space and action space will expand exponentially, thus reducing the convergence speed and taking up a lot of memory [72].

To solve this problem, **deep Q network (DQN)** [73] is proposed, which utilizes a DNN to approximate the Q-values. Compared with the classical RL algorithms, DQN has three advantages in dealing with EC with high complexity [74]. First, it is able to deal with high dimensional and complex systems. Second, it can learn the regularity of system environment. Last but not least, it is able to make optimal decisions based on current and past long-term reward. Therefore, some studies [75, 76] use DQN algorithms to optimize the control decision-making problems in EC and obtain good results.

However, DQN also has its shortcomings. Especially, when using nonlinear functions such as neural network to approximate the Q-function, the learning result of DRL is unstable or even divergent. To solve this problem, an experience replay mechanism using the prior experience is integrated into DQN [77, 78].

3.2.4 Federated Learning. **Federated learning (FL)** is a distributed ML framework, which can effectively help multiple organizations train models under the requirements of user privacy protection, data security, and government regulations [79]. In this framework, different local users do not need to put all the raw data on the central server for training, but train the local model through privacy related data, then all the local models are aggregated into a global model on the central server [80].

As discussed above, the goal of EC is to deploy computing tasks at the edge of the network near the client. However, the data of a single edge node may not meet the requirements of model training. Therefore, the cooperation model training between different nodes under data privacy protection is a research hotspot; see, e.g., Reference [81].

3.2.5 Evolutionary Algorithms. Evolutionary algorithms are a kind of optimization methods inspired by biological evolution mechanism and biological behavior [82]. Evolutionary algorithms include **particle swarm optimization (PSO)**, **genetic algorithm (GA)**, **differential evolution (DE)**, and so on.

Generally speaking, evolutionary algorithms are divided into the following steps. The first step is to initialize variables. After that, the evolutionary algorithms continuously iterate three steps named fitness evaluation and selection, population reproduction and variation, and population updating [82]. Finally, the second step is iterated until the termination condition is satisfied.

At present, evolutionary algorithm has been applied in many problems of EC, such as resource scheduling optimization [83], load balancing [84], and task scheduling [85]. In this article, we mainly discuss ML, a recently popular AI subclass, so evolutionary algorithm is only briefly introduced here.

3.3 Artificial Intelligence Solutions for Optimizing Edge Computing

Now, we are going to provide a comprehensive summary of studies (listed in Table 1) that uses AI methods to optimize EC in different scenarios including computing offloading, reducing energy consumption, increasing the security of EC, keeping data privacy, and resource allocation.

3.3.1 Computing Offloading Optimization. At present, more and more studies have begun to make full use of AI to solve computing offloading [86]. We will summarize the AI-based computing offloading schemes in existing research to reduce energy consumption, reduce latency, and reduce both.

Reducing energy consumption. In terms of reducing energy consumption, a partial computing offloading scheme based on DL decision-making is proposed by Ali et al. [31]. The authors establish a new type of decision-making process, which can intelligently select the optimal computing offloading strategy, thus reducing the total energy consumed in the execution of computing tasks. Compared with its previous work in Reference [87], this strategy additionally considers the energy consumption of user equipment in the cost function, which reduces its energy consumption by 3%.

Reducing latency. Although EC itself has the advantage of low latency compared to cloud computing, it still has room for optimization. Smart-Edge-CoCaCo [88] is proposed to minimize the latency by jointly optimizing the wireless communication model, the collaborative filter caching model, and the computing offloading model. In addition, since the computing power of edge devices is limited, offloading all tasks to edge devices may exceed the capacity of the edge device. With this in mind, Xu et al. [89] propose a DL-based heuristic offloading method. This method uses origin-destination electronic communications network distance estimation and heuristic searching to find the optimal computing offloading strategy.

Reducing both energy consumption and latency. All the methods mentioned in previous paragraphs either only minimize energy consumption, or only minimize latency. There are also studies that consider the minimization of both through RL. Kiran et al. [54] propose a scheme that uses Q-learning to make optimal control decisions to reduce the delay in EC and adds constraints to the cost function to reduce energy consumption in EC. Although this scheme has a good effect on reducing energy consumption and delay, it does not take into account the curse-of-dimensionality problem of EC.

Table 1. Summary of Research on AI-optimized EC

Problem	Goal	Citation	AI	Contribution
Computing offloading optimization	Reduce energy consumption	[98]	Distributed DL-based offloading algorithm	Add the cost of changing local execution tasks in the cost function
	Reduce latency	[88]	Smart-Edge-CoCaCo algorithm based on DL	Joint optimization of wireless communication, collaborative filter caching and computing offloading
		[89]	A heuristic offloading method	Origin-destination electronic communication network distance estimation and heuristic searching to find optimal strategy for shorting the transmission delay of DL tasks
	Reduce both energy consumption and latency	[54]	Cooperative Q-learning	Improve the search speed of traditional Q-learning
		[90]	TD learning with postdecision state and semi-gradient descent method	Approximate dynamic programming to cope with curse-of-dimensionality
		[91]	Online RL	Special structure of the state transitions to overcome curse-of-dimensionality; additionally consider the EC scenario with energy harvesting
		[93]	DRL-based offloading scheme	No prior knowledge of transmission delay and energy consumption model; compress the state space dimension through DRL to further improve the learning rate; additionally consider the EC scenario with energy harvesting
		[94]	DRL-based computing offloading approach	Markov decision process to represent computing offloading; learn network dynamics through DRL
		[95]	Q-function decomposition technique combined with double DQN	Double deep Q-network to obtain optimal computing offloading without prior knowledge; a new function approximator-based DNN model to deal with high dimensional state spaces
		[10]	RL based on neural network architectures	An infinite-horizon average-reward continuous-time Markov decision process to represent the optimal problem; a new value function approximator to deal with high dimensional state spaces

(Continued)

Table 1. Continued

Problem	Goal	Citation	AI	Contribution
Other ways to reduce energy consumption	Optimize the hardware structure of edge devices	[102]	Binary-weight CNN	A static random access memory for binary-weight CNN to reduce memory data throughput; parallel execution of CNN
		[104]	DNNs	FPGA-based binarized DNN accelerator for weed species classification
	Control device operating status	[105]	DRL-based joint mode selection and resource management approach	Reduce the medium- and long-term energy consumption by controlling the communication mode of the user equipment and the light-on state of the processors
	Combine with energy Internet	[106]	Model-based DRL	Solve the energy supply problem of the multi-access edge server
		[70]	RL	A fog-computing node powered by a renewable energy generator
Security of edge computing		[113]	Minimax-Q learning	Gradually learn the optimal strategy by increasing the spectral efficiency throughput
		[114]	Online learning	Reduced bandwidth usage by choosing the most reliable server
		[115]	Multiple AI algorithms	Algorithm selection mechanism capable of intelligently selecting optimal AI algorithm
		[117]	Hypergraph clustering	Improve the recognition rate by modeling the relationship between edge nodes and DDoS through hypergraph clustering
		[112]	Extreme Learning Machine	Show faster convergence speed and stronger generalization performance of the Extreme Learning Machine classifier than most classical algorithms
		[56]	Distributed DL	Reduce the burden of model training and improve the accuracy of the model
		[120]	DL, restricted Boltzmann machines	Give active learning capabilities to improve unknown attack recognition

(Continued)

Table 1. Continued

Problem	Goal	Citation	AI	Contribution
Privacy protection		[122]	Deep PDS-Learning	Speed up the training with additional information (e.g., the energy utilization of edge devices)
		[124]	Generative adversarial networks	An objective perturbation algorithm and an output perturbation algorithm that satisfy differential privacy
		[125]	A deep inference framework called EdgeSanitizer	Data can be used to the maximum extent, while ensuring privacy protection
Resource allocation optimization		[77]	Deep Q-learning	Derive trust values using uncertain reasoning; avoid local convergence by adjusting the learning rate
		[166]	Actor-critic RL	An additional DNN to represent a parameterized stochastic policy to further improve performance and convergence speed; a natural policy gradient method to avoid local convergence
		[76]	DRL-based resource allocation scheme	Additional SDN to improve QoS
		[127]	Multi-task DRL	Transform the last layer of DNN that estimates Q-function to support higher dimensional action spaces

The curse-of-dimensionality refers to the problem that the complexity of the problem solving will increase at an exponential speed as the dimensionality increases [90, 91]. To solve the curse-of-dimensionality problem, Xu et al. [91] propose an algorithm that uses the special structure of state transitions of the considered EC system to overcome the curse-of-dimensionality problem. It is worth noting that the authors use energy harvesting [92] to reduce the consumption of traditional energy by fully utilizing renewable energy, but the transmission delay model and the energy consumption model are required to be known (this requirement can be eliminated by the method proposed in Reference [93]).

Compared with RL algorithms, DRL algorithms have stronger abilities to deal with high-dimensional state space. Therefore, Cheng et al. [94] propose a model-free DRL-based computing offloading method based on a space-air-ground integrated network to reduce EC latency and energy consumption. This method uses Markov decision process to represent the computing offloading decision process, and uses DRL to learn network dynamics.

Yet the ability of DRL algorithms to cope with high-dimensional state space is not perfect in every respect. Chen et al. [95] propose a new DNN model based on function approximator, and they also adopt double deep Q-network so that the optimal offloading strategy can be discovered without prior knowledge. Similarly, Lei et al. [10] propose a new type of value function approximator to deal with high-dimensional state equations. The authors also use an infinite-horizon average-reward continuous-time Markov decision process to represent the optimal problem. Finally, DRL is applied to solve the optimal computing offloading decision to reduce the energy consumption and latency of EC.

The DRL-based methods mentioned above use a centralized style for model learning. However, there is a potential assumption in this style that edge devices in EC have sufficient computing power. In fact, many edge devices do not yet have such powerful computing capabilities. As a result, Ren et al. propose a distributed computing offloading strategy combining federated learning and multiple DRLs [96]. It is proved by experiments that this method outperforms the centralized learning method in reducing the transmission cost in EC. In addition, distributed learning also has the advantage of fast convergence [97]. This is proved in Reference [98] by the method of optimizing computing offloading through distributed ML.

3.3.2 Non-computation Offloading Methods to Reduce Energy Consumption. EC provides certain computing capabilities near the data source, so that many computing tasks do not need to be delivered to the cloud for execution. While this model brings high response speed to people, it will inevitably cause a surge in energy consumption on the edge side. Moreover, many applications in EC require AI algorithms to make real-time decisions (such as intelligent driving [99] and intelligent monitoring systems [100]), but AI algorithms are computationally intensive to varying degrees. This is a huge challenge for devices with limited power. From the perspective of overall energy consumption, with the gradual popularization and widespread application of AI, how to control global overall energy consumption or improve energy efficiency is also very important.

Apart from computation offloading, there are many other factors that affect the energy consumption of edge devices. For example, different AI algorithms and different hardware structures adopted by edge devices will also affect energy consumption [101]. We will introduce AI solutions to reduce EC energy consumption in terms of optimizing hardware structure, controlling operating status, and combining energy Internet.

Optimizing hardware structure. A **static random access memory (SRAM)** [102] is able to reduce memory data throughput, and it combines parallel CNNs to enable simultaneous access to different memory blocks. Experiments show that this architecture significantly reduces energy consumption compared to traditional digital accelerator using small bitwidths. Based on **field-programmable gate array (FPGA)** [103], Lammie et al. [104] design a binarized DNN accelerator for weed species classification, which reduces energy consumption by 7 times compared with GPU-based accelerator under the same conditions. The authors believe that well-cultivated FPGA-based accelerator for AI algorithms is an ideal choice for edge devices with limited resources but need to perform learning and reasoning tasks.

Controlling operating status. Sun et al. propose a method based on DRL to reduce the medium and long-term energy consumption of EC by controlling the communication modes of user devices and the light-on state of processors [105]. This method uses Markov process to model the energy consumption of cache states and cloud processors and DRL to make decisions. According to some constraints (quality of service constraints, transmission power constraints, and the computing capability constraint in the cloud), the method uses an iterative algorithm to optimize the precoding of user devices.

Combining Energy Internet. EC has distributed characteristics, and the workload of edge-side devices will dynamically change with different geographical locations and times, which makes the energy consumption of each edge node unpredictable and uneven. To deal with the huge energy demand of EC and its heterogeneity, the combination of energy Internet (including smart grid and microgrid) with EC can provide renewable energy for EC [70, 106]. Energy Internet is a distributed energy production model that achieves local energy self-sufficiency by making full use of renewable energy sources [107, 108]. This feature of energy Internet is very suitable for providing energy to EC, thereby reducing the consumption of non-renewable energy. Since renewable energy is infinite, reducing non-renewable energy consumption is also equivalent to reducing energy consumption. However, due to the uncertainty of renewable energy production [109], some studies [70, 106] also aim to balance the energy supply and demand of EC through DRL-based control strategies. With the deployment of EC devices into energy Internet, energy management will also become more complex [110]. DRL combined with curriculum learning [111] has been used to realize a bottom-up energy management scheme [110].

3.3.3 Security of Edge Computing. Delegating computing and storage tasks from the cloud to the edge can reduce the security problems caused by network congestion and centralization to some extent. However, the distributed environment of EC also brings new security problems, such as **distributed denial of service (DDoS)** attacks and jamming attacks that cause illegal distribution of distributed system resources [33, 112]. What was previously applicable to a centralized environment (like cloud computing) is no longer applicable to solving these new security issues. In this part, we will review the studies on improving the security of EC based on AI algorithms.

Traditional machine learning methods. Traditional ML can help with the identification and classification of different attacks. In response to jamming attacks that threaten EC security, Wang et al. [113] propose a stochastic game framework that maximizes the spectral efficiency throughput by minimax-Q learning, thereby gradually learning the optimal strategy. The disadvantage of this method is that it needs extra bandwidth to avoid jamming attacks. This can be avoided by selecting the most reliable server based on online learning to reduce the security risks caused by jamming attacks [114]. To reduce the false alarm rate and data transmission delay of traditional intrusion detection systems, an algorithm selection mechanism can be deployed on the edge side [115]. This enables intelligent selection of the optimal ML algorithm for edge devices to distinguish false alarms. The experimental results prove that the method based on AI algorithm can improve the security of EC more effectively than the method based on non-AI algorithm.

Among various network attacks, DDoS is a relatively common attack method. Hypergraph clustering [116] can be adopted to model the relationship between edge nodes and DDoS to improve the recognition rate [117]. Kozik et al. uses a single-layer neural network to build the extreme learning machine classifier [112]. In this method, the training task of the attack detection classifier model is performed in the cloud with powerful computing resources. The trained classifier model is then offloaded to the edge devices for attack detection. In addition, experiments have also proven that the extreme learning machine classifier has faster convergence speed and stronger generalization performance than most traditional classification algorithms (such as SVM, or single-layer perceptron).

DL methods. Although traditional ML algorithms can improve the accuracy and robustness of network attack detection and recognition, they lack the ability of automatic feature extraction [118]. As a result, traditional AI algorithms are not sensitive to known but slightly changed attacks. At the same time, due to the lack of prior knowledge of unknown vulnerabilities, they can not effectively detect zero-day attacks [119]. Deep learning, however, has been successfully applied in image processing, computer vision and many other fields in recent years because of its structure that can automatically mine and learn the hidden features in massive data [63]. Researchers begin to focus on DL, since the problem of cyber-security attack identification in EC is similar to the tasks in these fields.

Abeshu et al. [56] propose a DL-based method for attack detection in EC. To reduce the burden of model training and improve the accuracy of the model, this method uses a pretrained stacked autoencoder to screen the real valuable features and then uses softmax to do classification. This method shows great advantages in the aspects of availability, scalability and effectiveness compared with traditional ML algorithms. However, the authors fail to take into account the improvement of the detection rate of new attacks. This can be solved by unsupervised learning. The DL-based algorithm proposed in Reference [120] learns the characteristics of the attack through the deep belief network and uses the softmax function to identify various attacks on the EC. The difference is that this solution incorporates unsupervised learning restricted Boltzmann machines into the proposed model. Since unsupervised learning restricted Boltzmann machines is a stochastic artificial neural network with active learning characteristics, this model enables active learning to improve the recognition rate of attacks that have never occurred before.

3.3.4 Data Privacy. To a certain extent, EC reduces the risk of privacy leakage caused by uploading data to cloud servers that users cannot control. However, the problem of data privacy leakage also exists on the edge side. On the one hand, the distributed nature of EC brings new challenges to privacy protection. On the other hand, the application of AI on the edge side requires massive data for model training and reasoning, which are inevitably mixed with a large amount of user privacy. During the training process, some models may save part of the training set with private data, so an attacker can illegally obtain users' privacy by analyzing these models [121]. Consequently, it is very important to ensure the data privacy and security of edge-side users without affecting the performance of EC. This topic has attracted the attention of many researchers in recent years.

Post-decision state learning. A **post-decision state (PDS)** learning method is proposed in Reference [122], in which the state transition function is factored into known and unknown components. This method first uses the Markov decision process to describe EC's offloading problem and then solves the problem by combining PDS-learning technique with the traditional deep Q-network algorithm. This combination can well balance task scheduling and privacy protection. It is worth noting that compared with the traditional deep Q-network, the new algorithm can speed up the model training by learning some additional information (such as the energy utilization of edge devices).

Federated learning. A **privacy-preserving asynchronous FL mechanism (PAFLM)** for EC is proposed, which allows multiple edge nodes to realize more efficient FL without sharing private data and affecting inference accuracy [81]. Because the local model training of each node depends on the data inside the node to a large extent, it is easier to lead to local optimum. Through FL, the local model can be optimized with the help of the model parameters of other nodes, which can solve local optimum problem and improve the accuracy of model.

Differential privacy. To protect the user privacy in the training data set under EC, AI algorithms are usually combined with differential privacy, a system where including or excluding any piece of data will not change the results of related data analysis to a great extent [123]. In other words, by applying differential privacy, observers cannot tell from its output if any particular piece of information has been used [123]. Du et al. [124] propose two AI-based algorithms that satisfy differential privacy: *objective perturbation* algorithm and *output perturbation* algorithm. The difference between the two is that objective perturbation adds Laplace noise to objective functions, while output perturbation adds the noise to outputs. By injecting Laplace noise, ML algorithms show better efficiency and accuracy in prediction, and they are more effective in protecting the privacy of training data used in EC. Similarly, a deep reasoning framework based on differential privacy, called EdgeSanitizer, is proposed in Reference [125]. The framework uses as much useful information as possible with a DL-based data minimization method. Then it removes as much sensitive private information as possible from data sets by adding random noise to the original data through a local differential privacy method [126]. This approach ensures that the data is used to the maximum extent while protecting the privacy in EC.

3.3.5 Resource Allocation Optimization. DRL has been proven to be capable of handling dynamic decision problems with high-dimensional states and action spaces [127]. At present, some studies have focused on DRL to solve the resource allocation problem in EC.

The method in Reference [77] captures the fact that the EC environment state is constantly changing. The information about wireless channel conditions, each node's trust value, the contents in the cache, and the vacant computational capacity is passed to the DNN to estimate the Q-function. The network operator's revenue is regarded as the reward, and the agent trains the DNN through the obtained reward. It avoids local convergence by adjusting the learning rate. Although this method has a good effect, there is still room for improvement in convergence and performance.

Although the study above proves that DQN has a good performance in optimizing dynamic decision problems with high-dimensional state space, there are still some limitations when solving problems based on high-dimensional action space. Therefore, Chen et al. [127] propose a new DRL-based resource allocation decision framework that makes the following two contributions:

- The framework uses DNN to train with a self-supervised training process to predict the resource allocation action, with the training data generated by the **Monte Carlo tree search (MCTS)** [128] algorithm;
- The authors modify the last layer of the traditional DNN used to estimate Q-function, so that it can support higher-dimensional action space.

The experiment proves that compared with the method of directly using DQN, this method has reduced the delay by 51.71%.

3.4 Summary

In this section, we first explain the mutual benefit between AI and EC. Then, we introduce AI algorithms (especially traditional ML, DL, RL, and DRL) in detail. Finally, from the perspectives of task scheduling, resource allocation, privacy protection and security, the research results of using AI algorithms to optimize the performance of EC are reviewed. In the future, considering that the EC is faced with large-scale computing tasks, it would be very important to combine the multi-dimensional perspectives of network, computing, power allocation, and task scheduling for real-time joint optimization. To deal with these complex optimization problems, it is a potential research direction that uses the model-free method of AI algorithms to learn efficient strategies [11].

4 APPLICATION OF ARTIFICIAL INTELLIGENCE UNDER EDGE COMPUTING

In recent years, AI has made many achievements in various fields. Among them, smart city, smart manufacturing, and the IoV usually have more critical requirements for network delay and stability than other scenarios such as AR/VR, online gaming, or content distribution. Unfortunately, traditional cloud computing often fails to guarantee these requirements. Some researchers have started using EC to provide computing and storage resources on edge. To emphasize the advantages of EC in AI applications, this section will focus on summarizing the research results of AI applications in smart city, smart manufacturing, and the IoV under the EC framework.

This section summarize the existing research from the perspective of EC hierarchical architecture. The categorization of EC architecture, together with the corresponding target field and AI (ML) algorithm, are detailed in Table 2.

In this article, different EC architectures used in AI applications are summarized into three categories with detailed explanation and analysis. The three modes are: (a) the edge side is only

Table 2. Summary of AI Algorithms and Architectures

Field	Goal	DL	DRL	RL	Traditional ML	EC Architecture	Citation
Smart city	Security of city	√				(c)	[131]
		√				(c)	[100]
					√	(c)	[132]
	Urban healthcare	√				(b)	[133]
					√	(b)	[135]
					√	(c)	[51]
		√				(a)	[49]
	Urban energy management	√				(a)	[138]
			√			(b) & (c)	[140]
Smart manufacturing		√			√	(a)	[143]
					√	(b)	[50]
		√				(a)	[65]
		√				(b)	[145]
		√				(b)	[61]
Internet of Vehicles				√		(c)	[149]
		√				(c)	[152]
					√	(c)	[53]
		√		√		(b)	[153]
		√				(b)	[157]

The EC architectures are defined in Section 4, which can be divided into the following three categories. (a) The edge side is only responsible for data cleaning, and the cloud is responsible for training and reasoning. (b) The cloud is responsible for training, while the edge side is responsible for inference. (c) Delegate part or all of AI training and reasoning tasks to the edge (see Section 3.3.1 and Figure 4 for details).

responsible for data cleaning, and the cloud is responsible for training and reasoning; (b) the cloud is responsible for training, while the edge side is responsible for inference; (c) part or all of AI training and reasoning tasks are delegated to the edge (see Section 3.3.1 and Figure 4 for details). This section will accordingly summarize the research works (listed in Table 2) of AI application in many fields under above different EC hierarchical modes to emphasize the advantages of EC in AI application. Table 2 classifies and summarizes them from the perspective of architecture, AI algorithm, and target field.

4.1 Smart City

With the explosive growth of urban population and the trend of urbanization, the concept of smart city has been proposed and attracted widespread attention. Smart city uses smart means to reduce energy consumption in cities, enhance energy efficiency, ease traffic pressure [129], ensure the safety of cities and residents, and improve the quality of life of residents. In the smart city environment, there are a large number of hardware devices that generate data all the time. These devices include light smart devices for daily life (such as smart phones, smart bracelets, and portable medical devices), as well as surveillance cameras and various environmental detection sensors for urban security. AI is a good choice for smart city to improve the accuracy and efficacy of data analysis because of its proficiency in dealing with massive data [130].

In a population- and equipment-intensive area like a city, smart city has stricter requirements on real-time response and network stability to ensure the comfort and security of civil life in the city. However, the intensive computing tasks of AI training and reasoning pose a great challenge to the above requirements. To meet this challenge, some researchers have turned their attention to EC. We will subsequently describe in detail the schemes of using AI algorithms under EC architecture to deal with the problems in smart city scenarios.

4.1.1 Security of City. Smart cities need to continuously monitor the infrastructure and operation of the city, and they need to make quick judgments and respond quickly to security incidents. Integrating AI algorithms can improve the accuracy of security event identification. However, the network bandwidth is limited, and excessive data transmission will cause instability in network transmission. How to deal with massive data is therefore a very difficult problem for real-time monitoring systems. EC performs most of the data processing and analysis tasks on the edge and transmits only part of the data to the cloud. This can greatly reduce the network transmission pressure caused by massive monitoring data while improving the response speed of the application.

To ensure the safety of urban residents in public places or private places, a series of monitoring systems (e.g., traffic monitoring, indoor and outdoor monitoring, facility monitoring, violence and crime detection) need to be widely deployed to analyze and tackle the surrounding environment in real time. In urban monitoring, for instance, person re-identification is an important part to ensure the safety of residents. A new Siamese network architecture for person re-identification is proposed in Reference [131]. This architecture speeds up the retrieval of pedestrians by introducing EC. Considering that traditional methods may learn poorly and inefficiently due to the low resolution of images, together with the limited computing power on the edge side, the architecture introduces a residual model layer that can mine deep features and reduce the complexity of the global average pooling layer.

Utilizing the distributed characteristics of EC and the geo-distribution characteristics of monitoring data, it is a good idea to apply different AI algorithms to EC in a distributed way. A monitoring system based on distributed deep learning model is mentioned in Reference [100]. By introducing EC, the system reduces the cost of communication and improves response speed. This article uses the distributed characteristics of the edge side to deploy a distributed DL training method based on task-level and model-level parallel training. The goal is to speed up the training of the sub-model by taking advantage of different learning models while also using the computing power of edge nodes.

In contrast, Tang et al. [132] adopt the idea of configuring different AI algorithms in the edge and the cloud. The proposed general-purpose EC architecture for urban pipeline monitoring systems takes advantage of the low latency of edge nodes so that pipeline faults can be discovered in time, and response decisions can be made quickly. The architecture consists of four layers, and the architecture deploys different AI algorithms and control strategies in different layers to achieve low latency, low energy consumption, and high accuracy for smart pipeline monitoring to ensure the safety of pipelines in cities.

Challenges. In the process of protecting urban security, data privacy and security are also crucial. AI is an effective method of identifying malicious attacks and preventing privacy leakage, but the computing resources of edge devices are limited. Therefore, it is still a major challenge to design lightweight and effective AI algorithms suitable for EC [131].

4.1.2 Urban Healthcare. With the popularity of IoT and cloud computing, more and more personal medical devices are being used in daily life. These devices can collect users' physical data and upload the data to a cloud server. Through AI analysis, these data can greatly improve the accuracy of medical systems for disease classification and diagnosis. However, this model of cloud computing cannot really meet the requirements of telemedicine for time delay and data transmission.

Compared with traditional cloud computing, the application of EC meets the requirements of medical system for stable data transmission, transmission delay, and data security. In some emergency situations, for example, just the occurrence of errors such as long response time or data loss may directly threaten human life. Besides, EC has strong location awareness characteristics [33]. The higher processing speed of EC becomes a critical factor for location-sensitive medical systems.

Next, we will summarize existing urban medical and residents' health works that use EC to improve AI algorithms in terms of remote diagnosis and early warning of diseases, infectious disease prevention and control, and smart assessment.

Remote diagnosis and early warning. Muhammad et al. [133] propose a voice disorder assessment and treatment system. The sound data collected by the system is pre-processed by edge devices before being uploaded to the cloud. The system configures the CNN model to the edge server, so that the edge side has the capability of voice disorder detection and classification. Compared with the method without EC architecture in Reference [134], this method has lower latency and can effectively reduce the pressure on network bandwidth. However, this system still needs to send the diagnosis to a human expert, and the human expert decides the treatment plan.

For some diseases that are not easy to detect at an early stage and those that can be best treated in the early stages of the disease (e.g., lung cancer), the patient's survival can be significantly extended if a patient is diagnosed and treated early in the disease [135]. To improve the early diagnosis rate and accuracy of lung cancer, a lung cancer diagnosis system based on EC and AI is proposed in Reference [135]. This system can not only improve the early accuracy of lung cancer but also improve the efficiency and security of diagnosis. In the future, how to combine EC and AI algorithms to diagnose diseases and generate corresponding treatment plans without a human doctor is a valuable research direction.

Infectious disease prevention and control. The use of EC's powerful location awareness feature can effectively strengthen the prevention and control of infectious diseases. The healthcare framework proposed in Reference [51] can diagnose whether a user has been infected by Kyasanur forest disease and can map out areas where infectious diseases are likely to occur on the map. The network edge near the data source in this structure is responsible for data preprocessing, model training and reasoning. To more accurately identify infected people and outbreak-prone areas, this layer incorporates a classifier called EO-NN, which combines hybridization of the **extremal optimization (EO)** and the **neural networks (NN)**. Once a new infected person is detected, it will inform the infected person and nearby hospitals immediately. With the distributed nature of EC, the system has the ability to identify areas prone to infectious diseases.

Smart assessment. Residents' daily dietary structure management is also an important part of urban medical care, which also plays an important role in the prevention of diseases. Based on food image recognition, Liu et al. [49] propose a dietary assessment system under an EC architecture. The edge layer between end users and the cloud can minimize the response time and energy consumption, and the CNN algorithm can improve the accuracy of recognition. Compared to the previous system in Reference [136], which is only suitable for small data computing tasks, this system has the ability to perform large-scale data computing tasks.

Challenges. Medical diagnosis needs accurate judgment, which requires AI algorithms to extract all useful information from big data. However, the useful information that can be obtained by existing algorithms is rather limited. For supervised learning, manual labeling of data may also lead to unknown mistakes. In addition, the data acquisition system of smart medical in the future will be mainly deployed on wearable devices. To quickly analyze and respond to the collected data, it is also an important direction to deploy AI model to these wearable devices [136], which poses a great challenge to the energy supply of devices. How to balance the accuracy and lightweight of AI models is a direction worthy of studying [137].

4.1.3 Urban Energy Management. The trend of urbanization is also prompting the rapid increase of energy consumption in cities. This poses many challenges for urban energy management.

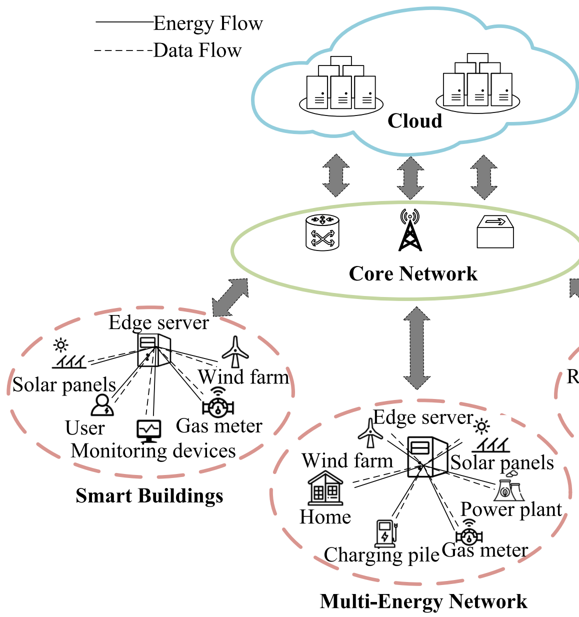

Fig. 5. A typical structure of smart energy management in smart city [140]. The architecture mainly includes three parts: (1) cloud with central control capability and powerful computing resources; (2) edge severs with local energy control through data analysis; (3) energy devices deployed at the terminal, including users, energy-producing and energy-consuming equipment, sensors, and so on.

For example, to meet the city's demand for energy, energy companies need to produce excess electricity to ensure continuous energy supply to the city. This leads to a certain degree of waste of energy [138]. In the era of big data, a large number of sensors deployed in various corners of the city can obtain data related to energy consumption in real time. These data include population density, electricity usage, and a wealth of environmental information that helps predict energy consumption and energy management. In addition, applying AI algorithm to energy management has greater advantages than traditional methods [139]. Under these conditions, the introduction of EC and AI can make energy consumption prediction and energy management faster and more accurate. A typical EC-based smart city energy management architecture is shown in Figure 5.

Real-time energy management decisions require dynamic predictions of energy consumption. However, the complexity and diversity of energy data and the dynamic nature of IoT data make it rather difficult to build an effective energy prediction system. In response to this problem, Liu et al. [140] design an EC-based energy management framework for reducing energy consumption in cities. Under this framework, the authors propose two DRL-based energy scheduling strategies:

- *Edge DRL*: model training and reasoning tasks are executed on the edge;
- *Cooperative DRL*: model training tasks are executed in the cloud, and dynamic energy management is implemented on the edge side based on models obtained from the cloud.

The authors prove by experiment that cloud-edge collaboration works best in terms of energy consumption, followed by the method of deploying AI algorithms only on the edge side, and the worst is the method of deploying AI algorithms only on the cloud [138]. This also indicates that EC is not a substitute for cloud computing, and the relationship between the two should be synergistic and complementary.

Challenges. The rapid growth of the number of edge devices deployed to cities has exacerbated the global energy crisis and global warming. One way to alleviate this problem is to use renewable energy to power edge devices. Considering that edge devices are scattered in different locations of the city, the energy consumption of traditional energy can be greatly reduced by using distributed renewable energy generation devices. However, this solution still faces many challenges, such as how to minimize the consumption of traditional energy while ensuring the normal operation of edge devices, and how to establish a complementary power system for different edge devices [140]. As a control center in EI system, energy router needs certain computing power [141, 142]. Therefore, it is also a feasible idea to combine energy router with EC in future research.

4.2 Smart Manufacturing

Introducing EC and AI in industrial production can maximize the use of hardware devices and the use of distributed computing and storage resources. The combination of the two also achieves efficient and secure resource management and task distribution, thereby greatly improving the plant's production efficiency, production quality and plant safety [143, 144].

Dynamic control. To improve the automation and intelligence of the real-time production control process, the authors of Reference [143] propose an intelligent robot factory system architecture called iRobot-Factory. With the assistance of EC, the architecture can dynamically adjust the configuration of the production line, collect and process a variety of data generated in the factory in real time, and identify and judge by AI means to achieve more efficient feedback control. The architecture shows great advantages over the traditional factory using cloud computing with respect to network communication time delay and recognition rate. Different devices in the factory need to cooperate with each other through groups to achieve swarm intelligence, not just each device operating independently. To realize swarm intelligence, how to use AI and EC technology in smart factory is a new challenge.

Equipment monitoring. In terms of industrial production site safety, it is essential to monitor the operating status of the machinery in the factory, since the quality issue of the machinery will inevitably arise during long-term work. To detect the running status of the machine, Wu et al. [50] propose an EC framework that includes a device layer, a local private edge cloud near the device layer, and a remote public cloud. The framework uses powerful public cloud to train the predictive model and then delegates the model to private edge cloud where online diagnostic and prognosis tasks are performed. This reduces the delay to a certain extent and enhances the accuracy of diagnosis and prognosis.

To better monitor and manage the equipment in the factory, it is important to clarify the type and quantity of onsite equipment. In response to the high cost of manual classification methods, a non-intrusive load monitoring system is proposed based on EC and LSTM [65]. In the system architecture, the edge is responsible for data cleaning and feature selection, while the cloud with the LSTM algorithm deployed analyzes power features uploaded by edge devices to classify and count field devices.

Defective product detection. In addition to ensuring the safety of factory equipment, some researchers have also turned their attention to monitor the quality of products more accurately and efficiently. Li et al. [145] build a DL-based product quality classification system for production quality monitoring, so that products with quality defects can be quickly detected on the edge side. The system deploys lower-level CNN layers at edge layers to capture defective products that are more easily to identify and high-level CNN in the cloud to capture defective products that are difficult to identify with edge layers. This design improves the efficiency and accuracy of identifying defective products, on the one hand, and it also reduces the network transmission cost, on the other hand.

Microseismic monitoring. In oil and gas production, the low signal-to-noise ratio and the need for real-time data transmission bring challenges in high-precision microseismic monitoring. Zhang et al. [61] design a neural network-based EC architecture called Edge-to-Center LearnReduce Microseismic Monitoring Platform under the environment of oil and gas production. The platform uses EC architecture with a new microseismic events detection algorithm based on LSTM, and CNN is deployed in the data center (i.e., the cloud). The model obtained through data training in the cloud will be delegated to each edge device, so that the edge device has the ability to recognize microseismic events. The real-time performance is improved by analyzing and processing data on the edge side that can get detection results faster and take corresponding actions. However, the data generated will first be processed by the edge device to extract useful information for the data center. This greatly reduces the volume of the data that need to transfer to the data center, so the platform can effectively improve transmission efficiency and reduce network transmission pressure. Experiments have shown that this monitoring platform combining neural network and EC can achieve an accuracy rate of more than 96% and improve the data transmission efficiency by about 90%.

4.3 Internet of Vehicles

IoV is currently a hot academic and commercial field, and it is a key step for humans to move towards an intelligent life in the future [147]. IoV can ease traffic congestion, reduce traffic accidents caused by improper driving, and improve passenger experience [99]. Abundant in-vehicle applications, road condition sensors, and intelligent systems bring a very convenient, comfortable, and safe riding experience for people traveling.

Although traditional cloud computing is currently the mainstream solution to the challenges brought by the increasing number of applications and data, it cannot meet the requirements of IoV (e.g., stable networks and low latency), due to the limitations of cloud computing itself. Using EC can effectively make up for the limitations of cloud computing [148]. IoV has the characteristics of limited resources, such as distributed computing and storage. How to allocate limited resources and how to schedule tasks are the problems that IoV needs to solve.

EC and AI can bring faster and more precise control, faster network communication, better user experience, and more computing resources for traditional vehicular network [149]. A typical EC-based IoV architecture is shown in Figure 6. Today, more and more fields use AI as a means to solve optimal strategies, and AI algorithms can also be applied to IoV to deal with the above problems. We will summarize the application of the combination of EC and AI in IoV from three perspectives: optimizing task offloading and resource allocation in IoV, improving the user experience of on-board entertainment, and improving vehicle intelligence.

4.3.1 Optimizing Task Offloading and Resource Allocation. The rapidly changing network structure, communication status, and computing load have led to the dynamics and uncertainty of task offloading [150], making efficient task offloading and resource allocation decisions more difficult. Feng et al. [148] use the ant colony optimization algorithm with fast convergence to solve the NP-hard task assignment problem. This method establishes multiple objective functions, and uses heuristics algorithm for optimization. However, this method is not good at making optimal decisions for offloading multiple data dependency tasks. In response to this problem, an EC framework for obtaining the optimal solution of task offloading through DRL is proposed in Reference [149]. The framework takes into account data dependencies, as well as resource requirements, vehicle movements, and access networks. It uses the asynchronous advantage **actor-critic (A3C)**

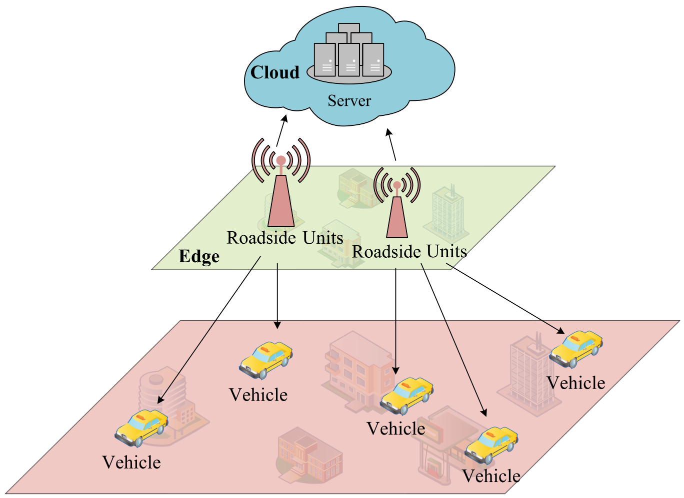

Fig. 6. A typical structure of IoV [146]. In this architecture, the edge is composed of roadside units with certain computing capabilities, so computing tasks on vehicles can be offloaded directly to roadside units for processing instead of offloading into the distant cloud [146].

algorithm [151] for the online optimization of task offloading decision to adapt to the dynamic changes of the vehicular network. Edge nodes will first distribute the trained decision model to the surrounding vehicles, and then upload the decision model online after vehicles' complete learning. To improve the performance of resource allocation and management, the prediction of wireless channel parameters is a very important means. Liu et al. [152] use LSTM to excel in spatio-temporal correlation in channel parameters and propose a wireless channel parameter prediction model based on LSTM and EC to optimize resource allocation and task scheduling in vehicular network.

In IoV, energy consumption is a huge obstacle that restricts its development. However, the studies mentioned above fail to consider the issue of energy consumption while making optimal offloading decisions. Yang et al. [53] put forward a joint optimization problem consisting of power control, user association, and resource allocation to minimize energy consumption in IoV. Finally, the feasible solution of this problem is obtained by an algorithm based on fuzzy c-means clustering that allows one data point to join multiple clusters.

4.3.2 Improving On-board Experience. The maturity and application of autonomous driving technology will bring more free time to passengers and drivers in the future. This will increase passengers and drivers' demand for on-board entertainment, such as listening to music, watching videos, and more [153]. These on-board entertainment activities have extremely high requirements for network latency, so implementing these computing-intensive applications in a connected vehicle with limited resources is facing great challenges [154]. These challenges include how to efficiently cache network content and how to efficiently schedule tasks and allocate resources.

The traditional content caching method is to cache the current popular content in roadside units in advance, but this also causes a waste of storage resources. To coordinate passenger experience and content caching costs, Hou et al. [153] propose a Q-learning-based caching strategy under the EC architecture. The action of this caching strategy consists of two parts, one is the cache amount, and the other is the roadside units to which the content is cached. The reward of this caching strategy is the elapsed time of transmitting the content required by the user. In addition, this article uses LSTM to predict the driving direction of the vehicle to better select roadside units.

In contrast, the method of Reference [155] imposes the task of content caching on both roadside units and vehicles. It uses a collaborative model based on Q-learning vehicles and roadside units for content caching and computation distribution. This model can make full use of the limited storage and computing resources of vehicles. In other words, the system will select vehicles and roadside units to perform the tasks of caching and computing according to the position and direction of motion of the car requesting the service. If the vehicles and roadside units around the car cannot meet their requirements, then the cache and calculation tasks will be handed over to the base station.

Aiming at the challenges of executing compute-intensive applications on cars with limited resources, Ning et al. [154] first use finite-state Markov chains to model vehicle-to-infrastructure communication and computing states and then express the resource allocation and task scheduling strategy as a goal to maximize users' **quality of experience (QoE)**.

4.3.3 Improving Vehicle Intelligence. In addition to the macro-control of resource allocation, it is also an important research direction to give AI technology to vehicle intelligence under the EC architecture [156]. For example, Ferdowsi et al. [157] propose an EC architecture that integrates DL to handle complex vehicle and traffic information. The architecture enables functions such as vehicle automatic control and driving route analysis. This architecture uses different DL algorithms according to the characteristics of different problems:

- Restricted Boltzmann machines are used to process complex data in **intelligent transportation systems (ITS)**;
- CNN and LSTM are used to perform real-time analysis of road conditions;
- Bi-RNN is used to predict driver behavior;
- LSTM is used to ensure data transmission security.

The increasing number of vehicles aggravates the problem of traffic jam. Traffic scheduling is a very effective way to deal with this problem. However, due to the large number of vehicles and the scale of road network, the number of routes that vehicles can choose increases exponentially. Therefore, it is not feasible to use centralized controller for route planning. Based on this problem, a distributed cooperative routing algorithm based on evolutionary game theory is proposed in Reference [158]. Each edge node deploys a **roadside unit (RSU)**, in which normal RSU is responsible for collecting traffic information, and game RSU controls nearby vehicles through proposed evolutionary game strategy.

4.3.4 Challenges. The combination of EC and IoV improves the response speed of vehicle scheduling and control, which further promotes the vehicle intelligence. However, there are still some challenges [159]. For example, when the vehicle is moving at a high speed, its communication connection needs to be switched between different edge servers, which may lead to a series of problems, such as disconnection or the degradation of user experience. In addition, one of the cores of IoV systems is resource sharing between different vehicles. As a result, how to set a reasonable incentive mechanism to encourage participants to share resources is vital. Finally, resource sharing will also bring some data privacy and security issues [160].

4.4 Summary

Table 2 summarizes the research works of combining EC with three different AI application scenarios. Apparently, these works adopt different AI algorithms and EC architectures in different scenarios according to their respective requirements for response speed, privacy, and so on, to maximize the performance of the AI models.

In essence, offloading all or part of the computing process of AI algorithms to the edge of the network is nevertheless to transfer AI computing tasks from a resource intensive environment to a resource limited environment [6]. Therefore, how to lighten AI models so that they can work efficiently at the edge of the network with limited computing, energy, and other resources needs further exploration [164]. In addition, an AI application often needs to collect data from different edge nodes, which poses a great threat to user privacy. Federated learning, as a very popular and potential research direction [96] can enable participants to learn jointly without sharing data. In recent years, the blockchain technology has been widely applied in many fields to establish mutual trust among participants in an open and distributed way [162, 165]. Incorporating blockchain to tackle the challenges of combined systems of AI and EC mentioned in this section is also a direction worthy of further exploration.

5 CONCLUSION

EC is a very promising new computing paradigm to make up for the shortcomings of existing cloud computing, while AI is a very popular field in both academia and industry. By summarizing the existing research results on the combination of AI and EC, we come to two conclusions. On the one hand, AI can further improve and optimize the performance of EC, because traditional non-AI methods have limitations in dealing with the complicated and dynamic environment in EC. On the other hand, EC can bring faster response time and more stable network status to the practical application of AI.

Although the research on combining AI and EC has made a lot of progress, there are still problems to be solved. For example, in the first aspect mentioned above, the complexity, dynamics, and high dimensions of the EC process make accurate modeling rather difficult. Therefore, it is an important research direction to design and adopt model-free methods to obtain efficient strategies [94]. In addition, for the second aspect, the key to deploying AI to the edge of the network is how to enhance the efficiency of AI algorithms with limited computing and energy resources, which requires further research and design of lightweight AI models [6, 164].

In summary, we hope that researchers will understand the importance of combining AI and EC and the mutually beneficial relationship between them through this article. We believe that there should be more academic research focusing on enabling EC to have higher computing offloading, privacy, and security performance and to enable wider use of AI. In the future, we plan to explore more research fields that combine the two, for example, distributed training and reasoning in the setting of EC.

REFERENCES

[1] A. U. R. Khan, M. Othman, S. A. Madani, and S. U. Khan. 2014. A survey of mobile cloud computing application models. *IEEE Commun. Surv. Tutor.* 16, 1 (2014), 393–413.

[2] F. Durao, F. Carvalho, A. Fonseka, and V. C. Garcia. 2014. A systematic review on cloud computing. *J. Supercomput.* 68, 3 (2014), 1321–1346.

[3] W. Shi and S. Dustdar. 2016. The promise of edge computing. *Computer* 49, 5 (2016), 78–81.

[4] M. Qin, L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei. 2018. Power-constrained edge computing with maximum processing capacity for IoT networks. *IEEE Internet Things J.* 6, 3 (2018), 4330–4343.

[5] A. M. Ghosh and K. Grolinger. 2021. Edge-cloud computing for internet of things data analytics: Embedding intelligence in the edge with deep learning. *IEEE Trans. Ind. Inform.* 17, 3 (2021), 2191–2200.

[6] P. Zhou, W. Chen, S. Ji, H. Jiang, L. Yu, and D. Wu. 2019. Privacy-preserving online task allocation in edge-computing-enabled massive crowdsensing. *IEEE Internet Things J.* 6, 5 (2019), 7773–7787.

[7] E. I. Gaura et al. 2013. Edge mining the internet of things. *IEEE Sens. J* 13, 10 (2013), 3816–3825.

[8] Z. Xu et al. 2020. Artificial intelligence for securing IoT services in edge computing: A survey. *Secur. Commun. Netw.* 2020 (2020), 1–13.

[9] C. Savaglio and G. Fortino. 2021. A simulation-driven methodology for IoT data mining based on edge computing. *ACM Trans. Internet. Techn.* 21, 2 (2021), 1–22.

[10] L. Lei, H. Xu, X. Xiong, K. Zheng, W. Xiang, and X. Wang. 2019. Multiuser resource control with deep reinforcement learning in IoT edge computing. *IEEE Internet Things J.* 6, 6 (2019), 10119–10133.

[11] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang. 2019. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proc. IEEE* 107, 8 (2019), 1738–1762.

[12] M. Miranda, C. Cristina, and S. Sebastiá. 2020. Deep learning at the mobile edge: Opportunities for 5G networks. *Appl. Sci.* 10, 14 (2020), 4735.

[13] F. Wang, M. Zhang, X. Wang, X. Ma , and J. Liu. 2020. Deep learning for edge computing applications: A state-of-the-art survey. *IEEE Access* 8 (2020), 58322–58336.

[14] J. Chen and X. Ran. 2019. Deep learning with edge computing: A review. *Proc. IEEE* 107, 8 (2019), 1655–1674.

[15] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen. 2020. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Commun. Surv. Tut.* 22, 2 (2020), 869–904.

[16] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief. 2019. Communication-efficient edge AI: Algorithms and systems. *IEEE Commun. Surv. Tut.* 22, 4 (2020), 2167–2191.

[17] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. 2013. Internet of things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comp. Syst.* 29, 7 (2013), 1645–1660.

[18] D. Reinsel, J. Gantz, and J. Rydning. 2018. Data age 2025. *The Digitization of the World: From Edge to Core.* IDC White Paper # US44413318.

[19] M. Marjani, F. Nasaruddin, A. Gani, I. Abaker, T. Hashem, A. Siddiqa, and I. Yaqoob. 2017. Big IoT data analytics: Architecture, opportunities, and open research challenges. *IEEE Access* 5 (2017), 5247–5261.

[20] Index, Cisco Global Cloud. Forecast and Methodology, 2016–2021 White Paper. Updated: February 1, 2018.

[21] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu. 2018. Data security and privacy-preserving in edge computing paradigm: Survey and open issues. *IEEE Access* 6 (2018), 18209–18237.

[22] S. Sukhmani, M. Sadeghi, M. Erol-Kantarci, and A. E. Saddik. 2019. Edge caching and computing in 5G for mobile AR/VR and tactile internet. *IEEE MultiMedia* 26, 1 (2019), 21–30.

[23] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos. 2017. IoT-based big data storage systems in cloud computing: perspectives and challenges. *IEEE Internet Things J.* 4, 1 (2017), 75–87.

[24] M. B. Mollah, M. A. K. Azad, and A. Vasilakos. 2017. Security and privacy challenges in mobile cloud computing: Survey and way ahead. *J. Netw. Comput. Appl.* 84 (2017), 38–54.

[25] Content Delivery Network. Retrieved from https://www.akamai.com/us/en/resources/content-delivery-network.jsp.

[26] M. Satyanarayanan. 2017. The emergence of edge computing. *Computer* 50, 1 (2017), 30–39.

[27] Brian E. Whitaker. 2019. Cloud edge computing: Beyond the data center. Retrieved from https://www.openstack.org/edge-computing/cloud-edge-computing-beyond-the-data-center/.

[28] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. 2016. Edge computing: Vision and challenges. *IEEE Internet Things J.* 3, 5 (2016), 637–646.

[29] R. Yang, F. Yu, P. Si, Z. Yang, and Y. Zhang. 2019. Integrated blockchain and edge computing systems: A survey, some research issues and challenges. *IEEE Commun. Surv. Tutor.* 21, 2 (2019), 1508–1532.

[30] L. Huang, X. Feng, A. Feng, Y. Huang, and L. Qian. 2018. Distributed deep learning-based offloading for mobile edge computing networks. *Mobile Netw. Appl.* (2018), 1–1. DOI: 10.1007/s11036-018-1177-x

[31] Z. Ali, L. Jiao, T. Baker, G. Abbas, Z. H. Abbas, and S. Khaf. 2019. A deep learning approach for energy efficient computational offloading in mobile edge computing. *IEEE Access* 7 (2019), 149623–149633.

[32] M. Yahuza, M. Idris, A. Wahid, A. T. S. Ho, S. Khan, N. Musa, and A. Taha. 2020. Systematic review on security and privacy requirements in edge computing: State of the art and future research opportunities. *IEEE Access* 8 (2020), 76541–76567.

[33] D. Liu, Z. Yan, W. Ding, and M. Atiquzzaman. 2019. A survey on secure data analytics in edge computing. *IEEE Internet Things J.* 6, 3 (2019), 4946–4967.

[34] G. Wang, X. Yang, W. Cai, and Y. Zhang. 2021. Event-triggered online energy flow control strategy for regional integrated energy system using Lyapunov optimization. *Int. J. Elec. Power* 125, 3 (2021), 106451.

[35] L. Chen, S. Zhou, and J. Xu. 2018. Computation peer offloading for energy-constrained mobile edge computing in small-cell networks. *IEEE ACM Trans. Netw.* 26, 4 (2018), 1619–1632.

[36] C. Liu, M. Bennis, M. Debbah, and H. V. Poor. 2019. Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing. *IEEE Trans. Commun.* 67, 6 (2019), 4132–4150.

[37] Y. Chiang, T. Zhang, and Y. Ji. 2019. Joint cotask-aware offloading and scheduling in mobile edge computing systems. *IEEE Access* 7 (2019), 105008–105018.

[38] M. Chen and Y. Hao. 2018. Task offloading for mobile edge computing in software defined ultra-dense network. *IEEE J. Select. Areas Commun.* 36, 3 (2018), 587–597.

[39] Z. Ning, P. Dong, X. Kong, and F. Xia. 2019. A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things. *IEEE Internet Things J.* 6, 3 (2019), 4804–4814.

[40] J. Du, L. Zhao, J. Feng, and X. Chu. 2018. Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee. *IEEE Trans. Commun.* 66, 4 (2018), 1594–1608.

[41] Y. Wang, X. Tao, X. Zhang, P. Zhang, and Y. H. Hou. 2019. Cooperative task offloading in three-tier mobile computing networks: An ADMM framework. *IEEE Trans. Veh. Technol.* 68, 3 (2019), 2763–2776.

[42] Z. Zheng, L. Song, Z. Han, G. Y. Li, and H. V. Poor. 2018. A stackelberg game approach to proactive caching in large-scale mobile edge networks. *IEEE Trans. Wirel. Commun.* 17, 8 (2018), 5198–5211.

[43] W. Jing, Q. Miao, H. Song, and X. Chen. 2019. Data loss and reconstruction of location differential privacy protection based on edge computing. *IEEE Access* 7, (2019), 75890–75900.

[44] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang. 2019. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. *IEEE Internet Things J.* 3, 6 (2019), 4660–4670.

[45] Q. Wang, D. Chen, N. Zhang, Z. Ding, and Z. Qin. 2017. PCP: A privacy-preserving content-based publish-subscribe scheme with differential privacy in fog computing. *IEEE Access* 5 (2017), 17962–17974.

[46] Y. Qiao, Z. Liu, H. Lv, M. Li, Z. Huang, Z. Li, and W. Liu. 2019. An effective data privacy protection algorithm based on differential privacy in edge computing. *IEEE Access* 7 (2019), 136203–136213.

[47] M. S. Hossain, G. Muhammad, and S. U. Amin. 2018. Improving consumer satisfaction in smart cities using edge computing and caching: A case study of date fruits classification. *Future Gener. Comp. Syst.* 88 (2018), 333–341.

[48] F. Samie, L. Bauer, and J. Henkel. 2019. From cloud down to things: An overview of machine learning in internet of things. *IEEE Internet Things J.* 6, 3 (2019), 4921–4934.

[49] C. Liu, Y. Cao, L. Yan, G. Chen, and H. Peng. 2018. A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure. *IEEE Trans. Serv. Comput.* 11 (2019), 249–261.

[50] D. Wu, S. Liu, L. Zhang, J. Terpenny, R. Gao, T. Kurfess, and J. Guzzo. 2017. A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing. *J. Manuf. Syst.* 43 (2017), 25–34.

[51] M. Abhishek, D. Tapas, et al. 2018. Kyasanur forest disease classification framework using novel extremal optimization tuned neural network in fog computing environment. *J. Med. Syst* 42, 10 (2018), 187.

[52] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya. 2020. Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet Things J.* 7, 8 (2020), 7457–7469.

[53] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei. 2019. Energy efficient resource allocation in UAV-enabled mobile edge computing networks. *IEEE Trans. Wirel. Commun.* 18, 9 (2019), 4576–4589.

[54] N. Kiran, C. Pan, S. Wang, and C. Yin. 2020. Joint resource allocation and computation offloading in mobile edge computing for SDN based wireless networks. *J. Commun. Netw.* 22, 1 (2020), 1–11. DOI: 10.1109/JCN.2019.000046

[55] Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, and C. Hsu. 2019. User allocation-aware edge cloud placement in mobile edge computing. *Software Pract. Exper.* 50, 10 (2019), 489–502.

[56] A. Abeshu and N. Chilamkurti. 2018. Deep learning: The frontier for distributed attack detection in fog-to-things computing. *IEEE Commun. Mag.* 56, 2 (2018), 169–175.

[57] Y. LeCun and Y. Bengio. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[58] D. L. Elliot. 1993. *A Better Activation Function for Artificial Neural Networks.* University of Maryland, Systems Research Center.

[59] H. Li, K. Ota, and M. Dong. 2018. Learning IoT in edge: Deep learning for the internet of things with edge computing. *IEEE Netw.* 32, 1 (2018), 96–101.

[60] P. Monkam, S. Qi, H. Ma, W. Gao, Y. Yao, and W. Qian. 2019. Detection and classification of pulmonary nodules using convolutional neural networks: A survey. *IEEE Access* 7 (2019), 78075–78091.

[61] X. Zhang, J. Lin, et al. 2018. An efficient neural-network-based microseismic monitoring platform for hydraulic fracture on an edge computing architecture. *Sensors* 18, 6 (2018), 1828.

[62] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9, 8 (1997), 1735–1780.

[63] A. Diro and N. Chilamkurti. 2016. Leveraging LSTM networks for attack detection in fog-to-things communications. *IEEE Commun. Mag.* 56, 9 (2016), 124–130.

[64] D. Park, S. Kim, Y. An, and J. Jung. 2018. iReD: A light-weight real-time fault detection system for edge computing using LSTM recurrent neural networks. *Sensors* 18 (2018), 2110–2124.

[65] C. Lai, W. Chen, L. Yang, and W. Qiang. 2019. LSTM and edge computing for big data feature recognition of industrial electrical equipment. *IEEE Trans. Ind. Inform.* 15 (2019), 2469–2477.

[66] B. Hussain, Q. Du, S. Zhang, A. Imran, and M. A. Imran. 2019. Mobile edge computing-based data-driven deep learning framework for anomaly detection. *IEEE Access* 7 (2019), 137656–137667.

[67] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic. 2019. Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin. *IEEE Trans. Wirel. Commun.* 18, 10 (2019), 4692–4707.

[68] S. A. Osia, A. S. Shamsabadi, A. Taheri, H. R. Rabiee, and H. Haddadi. 2018. Private and scalable personal data analytics using hybrid edge-to-cloud deep learning. *Computer* 51, 5 (2018), 42–49.

[69] Y. Wang, K. Wang, H. Huang, T. Miyazaki, and S. Guo. 2019. Traffic and computation co-offloading with reinforcement learning in fog computing for industrial applications. *IEEE Trans. Ind. Inf.* 15, 2 (2019), 976–986.

[70] S. Conti, G. Faraci, R. Nicolosi, S. A. Rizzo, and G. Schembra. 2017. Battery management in a green fog-computing node: A reinforcement-learning approach. *IEEE Access* 5 (2017), 21126–21138.

[71] X. Zhao, G. Huang, L. Gao, and M. Li. 2021. Low load DIDS task scheduling based on Q-learning in edge computing environment. *J. Netw. Comput. Appl.* 188, 1 (2021), 103095.

[72] B. Guo, X. Zhang, Y. Wang, and H. Yang. 2019. Deep-Q-network-based multimedia multi-service QoS optimization for mobile edge computing systems. *IEEE Access* 7 (2019), 160961–160972.

[73] V. Mnih, K. Kavukcuoglu, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[74] F. Xu, F. Yang, S. Bao, and C. Zhao. 2019. DQN inspired joint computing and caching resource allocation approach for software defined information-centric internet of things network. *IEEE Access* 7 (2019), 61987–61996.

[75] D. Zeng, L. Gu, S. Pan, J. Cai, and S. Guo. 2019. Resource management at the network edge: A deep reinforcement learning approach. *IEEE Netw.* 33, 3 (2019), 26–33.

[76] J. Wang, L. Zhao, J. Liu, and N. Kato. 2019. Smart resource allocation for mobile edge computing: A deep reinforcement learning approach. *IEEE Trans. Emerg. Top. Com.* DOI: 10.1109/TETC.2019.2902661

[77] Y. He, F. Yu, Y. He, S. Maharjan, and Y. Zhang. 2019. Secure social networks in 5G systems with mobile edge computing, caching, and device-to-device communications. *IEEE Wirel. Commun.* 25, 3 (2019), 103–109.

[78] Z. Qin, D. Liu, H. Hua, and J. Cao. 2021. Privacy preserving load control of residential microgrid via deep reinforcement learning. *IEEE Trans. Smart Grid* 12, 5 (2021), 4079–4089.

[79] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. Washington, USA. 1273–1282.

[80] S. Yu, X. Chen, Z. Zhou, X. Gong, and D. Wu. 2021. When deep reinforcement learning meets federated learning: Intelligent multitimescale resource management for multiaccess edge computing in 5G ultradense network. *IEEE Internet Things J.* 8, 4 (2021), 2238–2251.

[81] X. Lu, Y. Liao, P. Lio, and P. Hui. 2020. Privacy-preserving asynchronous federated learning mechanism for edge network computing. *IEEE Access* 8 (2020), 48970–48981.

[82] J. Zhang, Z. Zhan, Y. Lin, N. Chen, and Y. Gong. 2020. Evolutionary computation meets machine learning: A survey. *IEEE Comput. Intell. Mag.* 6, 4 (2020), 68–75.

[83] Y. Li and S. Wang. 2018. An energy-aware edge server placement algorithm in mobile edge computing. In *Proceedings of the IEEE International Conference on Edge Computing (EDGE'18)*. IEEE. San Francisco. 2018, 66–73.

[84] H. Gao, W. Li, R. Banez, Z. Han, and H. Poor. 2019. Mean field evolutionary dynamics in ultra dense mobile edge computing systems. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–6.

[85] C. Dong and W. Wen. 2019. Joint optimization for task offloading in edge computing: An evolutionary game approach. *Sensors* 19, 3 (2019), 740–764.

[86] Y. Zhan, S. Guo, P. Li, and J. Zhang. 2020. A deep reinforcement learning based offloading game in edge computing. *IEEE Trans. Comput.* 69, 6 (2020), 883–893.

[87] S. Yu, X. Wang, and R. Langar. 2017. Computation offloading for mobile edge computing: A deep learning approach. In *Proceedings of the IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC'17)*. IEEE, 1–6.

[88] Y. Hao, Y. Mian, L. Hu, M. S. Hossain, G. Muhammad, and S. U. Amin. 2019. Smart-edge-coCaCo: AI-enabled smart edge with joint computation, caching, and communication in heterogeneous IoT. *IEEE Netw.* 33, 2 (2019), 58–64.

[89] X. Xu, D. Li, Z. Dai, S. Li, and X. Chen. 2019. A heuristic offloading method for deep learning edge services in 5G networks. *IEEE Access* 7 (2019), 67734–67744.

[90] L. Lei, H. Xu, X. Xiong, K. Zheng, and W. Xiang. 2019. Joint computation offloading and multiuser scheduling using approximate dynamic programming in NB-IoT edge computing system. *IEEE Internet Things J.* 6, 3 (2019), 5345–5362.

[91] J. Xu, L. Chen, and S. Ren. 2017. Online learning for offloading and autoscaling in energy harvesting mobile edge computing. *IEEE Trans. Cogn. Commun. Netw.* 3, 3 (2017), 361–373.

[92] D. Mishra, S. De, S. Jana, S. Basagni, K. Chowdhury, and W. Heinzelman. 2015. Smart RF energy harvesting communications: Challenges and opportunities. *IEEE Commun. Mag.* 53, 4 (2015), 70–78.

[93] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang. 2019. Learning-based computation offloading for IoT devices with energy harvesting. *IEEE Trans. Veh. Technol.* 68, 2 (2019), 1930–1941.

[94] X. Cheng, L. Feng, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen. 2019. Space/aerial-assisted computing offloading for IoT applications: A learning-based approach. *IEEE J. Sel. Area. Comm.* 37, 5 (2019), 1117–1129.

[95] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis. 2019. Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning. *IEEE Internet Things J.* 6, 3 (2019), 4005–4018.

[96] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang. 2019. Federated learning-based computation offloading optimization in edge computing-supported internet of things. *IEEE Access* 7 (2019), 69194–69201.

[97] H. Cao and J. Cai. 2018. Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach. *IEEE Trans. Veh. Technol.* 67, 1 (2018), 752–764.

[98] L. Huang, X. Feng, A. Feng, Y. Huang, and L. Qian. 2018. Distributed deep learning-based offloading for mobile edge computing networks. *Mobile Netw. Appl.* 66, 12 (2018), 6353–6367.

[99] Z. Ning, P. Dong, X. Wang, L. Guo, and R. Kwok. 2019. Deep reinforcement learning for intelligent internet of vehicles: An energy-efficient computational offloading scheme. *IEEE Trans. Cogn. Commun.* 5 (2019), 1060–1072.

[100] J. Chen, K. Li, Q. Deng, K. Li, and P. Yu. 2019. Distributed deep learning model for intelligent video surveillance systems with edge computing. *IEEE Trans. Ind. Inform.* 5, 4 (2019), 1060–1072.

[101] S. A. Magid, F. Petrini, and B. Dezfouli. 2019. Image classification on IoT edge devices: Profiling and modeling. *Cluster Comput.* 23 (2019), 1025–1043.

[102] A. Biswas and A. P. Chandrakasan. 2019. CONV-SRAM: An energy-efficient SRAM with in-memory dot-product computation for low-power convolutional neural networks. *IEEE J. Solid-St. Circ.* 54, 1 (2019), 217–230.

[103] C. Maxfield. 2004. The design warrior's guide to FPGAs: Devices, tools and Flows. Elsevier.

[104] C. Lammie, A. Olsen, T. Carrick, and M. Rahimi Azghadi. 2019. Low-power and high-speed deep FPGA inference engines for weed classification at the edge. *IEEE Access* 7 (2019), 51171–51184.

[105] Y. Sun, M. Peng, and S. Mao. 2019. Deep reinforcement learning-based mode selection and resource management for green fog radio access networks. *IEEE Internet Things J.* 6, 2 (2019), 1960–1971.

[106] M. S. Munir, S. F. Abedin, N. H. Tran, and C. S. Hong. 2019. When edge computing meets microgrid: A deep reinforcement learning approach. *IEEE Internet Things J.* 6, 5 (2019), 7360–7374.

[107] H. Hua, Y. Qin, C. Hao, and J. Cao. 2018. Stochastic optimal control for energy internet: A bottom-up energy management approach. *IEEE Trans. Ind. Inf.* 15, 3 (2019), 1788–1797.

[108] Y. Qin, H. Hua, and J. Cao. 2019. Stochastic optimal control scheme for battery lifetime extension in islanded microgrid. *IEEE Trans. Smart Grid* 10, 4 (2019), 4467–4475.

[109] H. Hua, J, Cao, G. Yang, and R. Guang. 2018. Voltage control for uncertain stochastic nonlinear system with application to energy internet: Non-fragile robust H_∞ approach. *J. Math. Anal. Appl.* 463, 1 (2018), 93–110.

[110] H. Hua, Z. Qin, N. Dong, M. Ye, Z. Wang, X. Chen, and J. Cao. 2022. Data-driven dynamical control for bottom-up energy internet system. *IEEE Trans. Sustain. Energ.* 13, 4, (2022), 315–327.

[111] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. 2009. Curriculum learning. In *Proceedings of the International Conference on Machine Learning (ICML'09)*. ACM. Montreal, 2009, 41–48.

[112] R. Kozik, M. Ficco, M. Choraś, and F. Palmieri. 2018. A scalable distributed machine learning approach for attack detection in edge computing environments. *J. Parallel Distr. Com.* 119 (2018), 18–26.

[113] B. Wang, Y. Wu, K. R. Liu, and T. C. Clancy. 2011. An anti-jamming stochastic game for cognitive radio networks. *IEEE J. Select. Area. Commun.* 29, 4 (2011), 877–889.

[114] B. Li, T. Chen, and G. B. Giannakis. 2019. Secure mobile edge computing in IoT via collaborative online learning. *IEEE Trans. Signal Process.* 67, 23 (2019), 5922–5935.

[115] Y. Wang, W. Meng, W. Li, Z. Liu, Y. Liu, and H. Xue. 2019. Adaptive machine learning-based alarm reduction via edge computing for distributed intrusion detection systems. *Concurr. Comp.-Pract. E* 31, 19 (2019), 1–12.

[116] L. Yu, X. Shen, J. Yang, K. Wei, and R. Xiang. 2020. Hypergraph clustering based on game-theory for mining microbial high-order interaction module. *Evolution. Bioinform. Online* 16 (2020), 117693432097057.

[117] X. An, J. Su, X. Lu, and F. Lin. 2018. Hypergraph clustering model-based association analysis of DDOS attacks in fog computing intrusion detection system. *J. Wirel. Comm. Netw.* 1 (2018), 249–258.

[118] L. Fernández Maimó, A. Huertas Celdrán, M. Gil Pérez, F. García Clemente, and G. Martínez Pérez. 2019. Dynamic management of a deep learning-based anomaly detection system for 5G networks. *J. Amb. Intel. Hum. Comp.* 10, 8 (2019), 3083–3097.

[119] M. Zhang, L. Wang, S. Jajodia, A. Singhal, and A. Massimiliano. 2016. Network diversity: A security metric for evaluating the resilience of networks against zero-day attacks. *IEEE Trans. Inf. Foren.* Section 11, 5 (2016), 1071–1086.

[120] Y. Chen, Y. Zhang, S. Maharjan, M. Alam, and T. Wu. 2019. Deep learning for secure mobile edge computing in cyber-physical transportation systems. *IEEE Netw.* 33, 4 (2019), 36–41.

[121] M. Du, K. Wang, Y. Chen, X. Wang, and Y. Sun. 2018. Big data privacy preserving in multi-access edge computing for heterogeneous internet of things. *IEEE Commun. Mag.* 56, 8 (2018), 62–67.

[122] X. He, R. Jin, and H. Dai. 2019. Deep PDS-learning for privacy-aware offloading in MEC-enabled IoT. *IEEE Internet Things J.* 6, 3 (2019), 4547–4555.

[123] T. Wang, H. Hua, Z. Wei, and J. Cao. 2022. Challenges of blockchain in new generation energy systems and future outlooks. *Int. J. Elec. Power* 135, 107499 (2022), 265–284.
[124] M. Du, K. Wang, Z. Xia, and Y. Zhang. 2020. Differential privacy preserving of training model in wireless big data with edge computing. *IEEE Trans. Big Data* 6, 2 (2020), 283–295.
[125] C. Xu, J. Ren, L. She, Y. Zhang, Z. Qin, and K. Ren. 2019. EdgeSanitizer: Locally differentially private deep inference at the edge for mobile data analytics. *IEEE Internet Things J.* 6 (2019), 5140–5151.
[126] C. Dwork and A. Roth. 2014. The algorithmic foundations of differential privacy. *Found. Trends. Theor. Comput. Sci.* 9, 3 (2014), 211–407.
[127] J. Chen, S. Chen, Q. Wang, B. Cao, G. Feng, and J. Hu. 2019. iRAF: A deep reinforcement learning approach for collaborative mobile edge computing IoT networks. *IEEE Internet Things J.* 6, 4 (2019), 7011–7024.
[128] G. Chaslot, S. Bakkes, I. Szita, and P. Spronck. 2008. Monte-carlo tree search: A new framework for game AI. In *Proceedings of the 4th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. 216–217.
[129] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei. 2021. An edge traffic flow detection scheme based on deep learning in an intelligent transportation system. *IEEE Trans. Intell. Transp.* 22, 3 (2021), 1840–1852.
[130] I. Hashem, V. Chang, et al. 2016. The role of big data in smart city. *Int. J. Inform. Manage.* 36, 5 (2016), 748–758.
[131] S. Pang, S. Qiao, T. Song, J. Zhao, and P. Zheng. 2019. An improved convolutional network architecture based on residual modeling for person re-identification in edge computing. *IEEE Access* 7 (2019), 106749–106760.
[132] B. Tang, Z. Chen, G. Hefferman, S. Pei, W. Tao, H. He, and Q. Yang. 2017. Incorporating intelligence in fog computing for big data analysis in smart cities. *IEEE Trans. Ind. Inform.* 13, 5 (2017), 2140–2150.
[133] G. Muhammad, M. Alhamid, M. Alsulaiman, and B. Gupta. 2018. Edge computing with cloud for voice disorder assessment and treatment. *IEEE Commun. Mag.* 56, 4 (2018), 60–65.
[134] G. Muhammad, M. Rahman, A. Alelaiwi, and A. Alamri. 2017. Smart health solution integrating IoT and cloud: A case study of voice pathology monitoring. *IEEE Commun. Mag.* 55, 1 (2017), 69–73.
[135] M. Prabukumar, L. Agilandeeswari, and K. Ganesan. 2017. An intelligent lung cancer diagnosis system using cuckoo search optimization and support vector machine classifier. *J. Amb. Intel. Hum. Comp.* 3 (2017), 1–27.
[136] V. Stantchev, A. Barnawi, S. Ghulam, and J. Schubert. 2015. Smart items, fog and cloud computing as enablers of servitization in healthcare. *Sensors Transduc.* 185 (2015), 121–128.
[137] J. Zhang and D. Tao. 2020. Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet Things J.* 8, 10 (2020), 7789–7817.
[138] H. Luo, H. Cai, Y. Sun, and L. Jiang. 2019. A short-term energy prediction system based on edge computing for smart city. *Future Gener. Comp. Syst.* 101 (2019), 444–457.
[139] H. Hua, Z. Wei, Y. Qin, T. Wang, and J. Cao. 2021. A review of distributed control and optimization in energy Internet: From traditional methods to artificial intelligence-based methods, *IET Cy-Phys. Syst.: Theory Appl.* 6, 2 (2021), 63–79.
[140] Y. Liu, C. Yang, et al. 2019. Intelligent edge computing for IoT-based energy management in smart cities. *IEEE Netw.* 33, 2 (2019), 111–117.
[141] C. Hao, Y. Qin, and H. Hua. 2020. Energy "routers," "computers," and "protocols." In *Energy Internet: Systems and Applications*. Springer Nature Switzerland AG, 193–208.
[142] H. Liang, H. Hua, Y. Qin, M. Ye, S. Zhang, and J. Cao. 2022. Stochastic optimal energy storage management for energy routers via compressive sensing. *IEEE Trans. Ind. Inform.* 18, 4 (2022) 2192–2202.
[143] L. Hu, Y. Miao, G. Wu, M. Hassan, and I. Humar. 2018. IRobot-factory: An intelligent robot factory based on cognitive manufacturing and edge computing, future gener. *Comp. Syst.* 90, 10 (2018), 1–13.
[144] F. Liang, W. Yu, X. Liu, D. Griffith, and N. Golmie. 2020. Toward edge-based deep learning in industrial internet of things. *IEEE Internet Things J.* 7, 5 (2020), 4329–4341.
[145] L. Li, K. Ota, and M. Dong. 2018. Deep learning for smart industry: Efficient manufacture inspection system with fog computing. *IEEE Trans. Ind. Inform.* 14 (2018), 4665–4673.
[146] Z. Ning, P. Dong, X. Wang, L. Guo, and R. Kwok. 2019. Deep reinforcement learning for intelligent internet of vehicles: An energy-efficient computational offloading scheme. *IEEE T. Cogn. Commun. Netw.* 5, 4 (2019), 1060–1072.
[147] L. Guo, M. Dong, Z. Chen, S. Feng, and G. Fang. 2017. A secure mechanism for big data collection in large scale internet of vehicle. *IEEE Internet Things J.* 4, 2 (2017), 601–610.
[148] J. Feng, Z. Liu, C. Wu, and Y. Ji. 2017. AVE: Autonomous vehicular edge computing framework with ACO-based scheduling. *IEEE Trans. Veh. Technol.* 66, 12 (2017), 10660–10675.
[149] Q. Qi, J. Wang, Z. Ma, H. Sun, Y. Cao, L. Zhang, and J. Liao. 2019. Knowledge-driven service offloading decision for vehicular edge computing: A deep reinforcement learning approach. *IEEE Trans. Veh. Technol.* 68 (2019), 4192–4203.
[150] Y. Sun, X. Guo, J. Song, S. Zhou, Z. Jiang, X. Liu, and Z. Niu. 2019. Adaptive learning-based task offloading for vehicular edge computing systems. *IEEE Trans. Veh. Technol.* 68 (2019), 3061–3074.
[151] H. Hua, Y. Qin, C. Hao, and J. Cao. 2019. Optimal energy management strategies for energy internet via deep reinforcement learning approach. *Appl. Energ.* 239 (2019), 598–609.

[152] G. Liu, Y. Xu, Z. He, Y. Rao, J. Xia, and L. Fan. 2019. Deep learning-based channel prediction for edge computing networks toward intelligent connected vehicles. *IEEE Access* 7 (2019), 114487–114495.

[153] L. Hou, L. Lei L, Z. Kan, and X. Wang. 2018. A Q-learning based proactive caching strategy for non-safety related services in vehicular networks. *IEEE Internet Things J.* 6 (2018), 4512–4520.

[154] Z. Ning, P. Dong, J. Rodrigues, and Z. Ning. 2019. Deep reinforcement learning for vehicular edge computing: an intelligent offloading system. *ACM Trans. Intel. Syst. Tec.* 10, 6 (2019), 1–1.

[155] T. Le and R. Hu. 2018. Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning. *IEEE Trans. Veh. Technol.* 67 (2018), 10190–10203.

[156] M. Khayyat, I. A. Elgendy, A. Muthanna, A. Alshahrani, S. Alharbi, and A. Koucheryavy. 2020. Advanced deep learning-based computational offloading for multilevel vehicular edge-cloud computing networks. *IEEE Access* 8 (2020), 137052–137062.

[157] A. Ferdowsi, U. Challita, and W. Saad. 2018. Deep learning for reliable mobile edge analytics in intelligent transportation systems. *IEEE Veh. Technol. Mag.* 14, 1 (2018), 62–70.

[158] J. Lu, J. Li, Yuan Q, and B. Chen. 2019. A multi-vehicle cooperative routing method based on evolutionary game theory. In *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC'19)*. IEEE. New Zealand. 987–994.

[159] X. Hou, Z. Rem, J. Wang, W. Cheng, and H. Zhang. 2020. Reliable computation offloading for edge-computing-enabled software-defined IoV. *IEEE Internet Things J.* 7, 8 (2020), 7097–7111.

[160] J. Zhang and K. B. Letaief. 2020. Mobile edge intelligence and computing for the internet of vehicles. *Proc. IEEE* 108, 2 (2020), 246–261.

[161] X. Wang, Y. Han, V. Leung, D. Niyato, X. Yan, and X. Chen. 2019. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Commun. Surv. Tut.* 22, 2 (2019), 869–904.

[162] M. KKowalski, Z. Lee, and T. Chan. 2021. Blockchain technology and trust relationships in trade finance. *Technol. Forecast. Soc.* 166 (2020), 120641.

[163] T. Wang, J. Guo, et al. 2021. RBT: A distributed reputation system for blockchain-based peer-to-peer energy trading with fairness consideration. *Appl. Energ.* 295, 1 (2021), 117056.

[164] S. Pang, S. Qiao, T. Song, J. Zhao, and P. Zheng. 2019. An improved convolutional network architecture based on residual modeling for person re-identification in edge computing. *IEEE Access* 7 (2019), 106748–106759.

[165] J. Li, J. Wu, J. Li, A. K. Bashir, M. J. Piran, and A. Anjum. 2021. Blockchain-based trust edge knowledge inference of multi-robot systems for collaborative tasks. *IEEE Commun. Mag.* 59, 7 (2021), 94–100.

[166] Y. Wei, F. Yu, M. Song, and Z. Han. 2019. Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor-critic deep reinforcement learning. *IEEE Internet Things J.* 6, 2 (2019), 2061–2073.

Received September 2021; revised June 2022; accepted August 2022

AUTHOR QUERIES

Q1: AU: Please check that you have provided each author's complete mailing and email address.
Q2: AU: Please check changes to Reference 54.