IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received February 20, 2020, accepted March 13, 2020, date of publication March 23, 2020, date of current version April 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982411

Deep Learning for Edge Computing Applications: A State-of-the-Art Survey

FANGXIN WANG[1], **(Student Member, IEEE), MIAO ZHANG**[1], **XIANGXIANG WANG**[1], **XIAOQIANG MA**[2], **AND JIANGCHUAN LIU**[1], **(Fellow, IEEE)**

[1]School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada
[2]School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China

Corresponding author: Xiaoqiang Ma (mxqhust@gmail.com)

This work was supported in part by the Canada NSERC Discovery Grant and in part by the National Natural Science Foundation of China under Grant 61702204.

ABSTRACT With the booming development of Internet-of-Things (IoT) and communication technologies such as 5G, our future world is envisioned as an interconnected entity where billions of devices will provide uninterrupted service to our daily lives and the industry. Meanwhile, these devices will generate massive amounts of valuable data at the network edge, calling for not only instant data processing but also intelligent data analysis in order to fully unleash the potential of the edge big data. Both the traditional cloud computing and on-device computing cannot sufficiently address this problem due to the high latency and the limited computation capacity, respectively. Fortunately, the emerging edge computing sheds a light on the issue by pushing the data processing from the remote network core to the local network edge, remarkably reducing the latency and improving the efficiency. Besides, the recent breakthroughs in deep learning have greatly facilitated the data processing capacity, enabling a thrilling development of novel applications, such as video surveillance and autonomous driving. The convergence of edge computing and deep learning is believed to bring new possibilities to both interdisciplinary researches and industrial applications. In this article, we provide a comprehensive survey of the latest efforts on the deep-learning-enabled edge computing applications and particularly offer insights on how to leverage the deep learning advances to facilitate edge applications from four domains, i.e., smart multimedia, smart transportation, smart city, and smart industry. We also highlight the key research challenges and promising research directions therein. We believe this survey will inspire more researches and contributions in this promising field.

INDEX TERMS Internet of Things, edge computing, deep learning, intelligent edge applications.

I. INTRODUCTION

With the explosive development of the Internet-of-Things (IoT) as well as the communication technologies such as WiFi and 5G, our future world is envisioned as an interconnected entity where billions of digital devices would provide uninterrupted services to both our daily lives and the industry. As reported by Cisco [1], there will be more than 50 billion IoT devices connected by the Internet by 2020. Such numerous IoT devices will generate a myriad of valuable data which, once being well processed effectively and efficiently, can empower many groundbreaking applications. Traditional computing architecture relies on cloud computing to provide sufficient computation capacity and sustainable energy. In this system, IoT devices are responsible to collect the data and deliver it to the remote powerful cloud, and the cloud servers will carry out the computation-intensive tasks and distributed the result back. However, the large latency caused by the long physical distance can sometimes become unacceptable, especially for those latency-sensitive applications like autonomous driving and highly interactive applications such as VR gaming. In addition, the huge data communication also greatly increases the pressure of the backbone network, bringing large overhead and cost to service providers.

The emerging edge computing [2] provides a promising solution for this problem. Though with many representation forms, such as fog computing [3] and cloudlet [4], the basic idea of edge computing is that the computation capacity should be deployed close to the data source for data processing, rather than transmitting the data to places with

The associate editor coordinating the review of this manuscript and approving it for publication was Xu Chen.

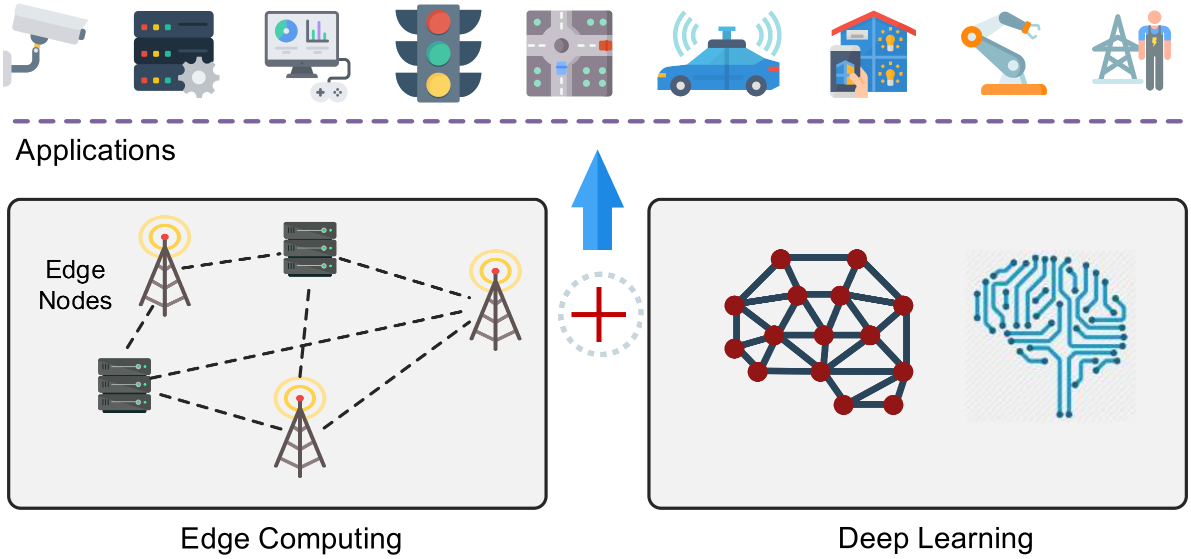

FIGURE 1. **The illustration of deep learning enabled edge computing applications.**

computation power. In this way, massive numbers of servers are deployed at the edge of the network and the tasks at IoT end devices can be offloaded to the edge servers for instant processing. The paradigm of edge computing brings many benefits compared to cloud computing. First, since data computing happens closer to the data source, the communication latency can be largely reduced, facilitating the development of latency-sensitive applications. Besides, local computation can better protect data privacy and application security. Last but not least, data processing at the network edge can effectively reduce traffic at the backbone network so as to alleviate the network pressure.

Deep learning has made remarkable breakthroughs in recent years due to the powerful perception ability. It has been widely used in various fields such as computer vision [5] and natural language processing [6]. Besides, its performance in computer and chess games, e.g., Atari Games [7] and the game of Go [8], even exceeds the best level of human players. The confluence of edge computing and deep learning will undoubtedly sheds a light on address the current challenges, enabling more desirable applications. On one hand, the applications of edge computing urgently need the powerful processing capabilities of deep learning to handle various complicated scenarios, such as video analytics [9], transportation control [10], etc. On the other hand, edge computing has provided specifically designed hardware foundations and platforms to better support deep learning running at the edge, e.g., the light-weighted Nvidia Jetson TX2 developing kit.[1] Though lots of pioneer efforts have been made towards deep-learning-enabled edge computing applications, this field is still in the infant stage.

Several existing surveys have investigated the convergence of deep learning and edge computing in the literature. Han *et al.* [11] presented their understanding on edge computing and deep learning from five aspects, while they did not make a comprehensive and in-depth overview from the perspective of applications. Similarly, Chen and Ran [12] focused on multiple aspects in deep learning and edge computing, but only mentioned a general abstraction for those emerging applications. Zhou *et al.* [13] mainly focused on the deep learning model training and inference with edge computing. There are also a series of surveys for mobile edge computing [2], [14]–[16] and deep learning [17], [18], respectively, while they focused on either of them without a comprehensive review on the combination. Therefore, a complete survey on the current cutting-edge researches is required at this time to provide a comprehensive review on deep-learning-enabled edge computing applications and illuminate the potential future directions.

To fulfill this gap, in this article, we focus on the confluence of edge computing and deep learning, and conduct an up-to-date literature review on the latest advances of leveraging deep learning to empower the edge computing applications, as illustrated in Fig. 1. We first provide a brief overview of edge computing and deep learning on concepts, advantages as well as representative technologies. We then summarize the deep-learning-enabled edge computing applications into four representative domains, i.e., smart multimedia, smart transportation, smart city and smart industry, which cover a series of crucial applications like video analytics, autonomous driving, intelligent traffic control, industrial manufacturing, etc. At last, we discuss some key research challenges and promising research directions to achieve stable, robust, and practical edge learning applications. Different from existing surveys, this article focused on the deep

[1] https://developer.nvidia.com/embedded/jetson-tx2

learning enabled edge computing applications, presenting a comprehensive review and highlighting the challenges and opportunities.

The rest of this article is organized as follows. We present the basic paradigm understanding of edge computing and its advantages in section II. We introduce some deep learning techniques in section III. The review of deep-learning-enabled edge applications is summarized in section IV. We highlight the challenges and research directions in section V. We at last conclude this article in section VI.

II. EDGE COMPUTING OVERVIEW

The emerging edge computing in recent years has seen successful development in various fields given its great potential in reducing latency and saving cost. Different from the cloud computing architecture, edge computing enables data processing at the edge of the network. On one hand, data computing is put closer to the data source, which greatly facilitates the development of delay-sensitive applications. On the other hand, the network traffic is largely reduced since the local processing avoids much data transmission, which remarkably saves the cost. In this section, we briefly introduce some edge computing paradigms and highlight the key advantages of edge computing.

A. EDGE COMPUTING RELATED PARADIGMS

The key component in edge computing is the edge devices, which are usually edge servers located closer at the network end for data processing, communication, caching, etc. There are also some other paradigms or technologies that share similar concepts with edge computing. We next discuss and differentiate some typical paradigms that are related to edge computing.

1) CLOUDLET

Cloudlet, initiated by Carnegie Mellon University, is envisioned as small clusters with certain computation and storage capabilities deployed near the mobile devices such as buildings and shopping centers for assisted processing, offloading, caching, etc. Cloudlet usually utilizes virtualization management technologies [4] to better support mobile applications. And an important target of cloudlet is to bring the cloud advances to mobile users [19], achieving more low-latency and resourceful processing. Micro data centers (MDCs) [20], initiated by Microsoft that are similar to the concept of cloudlet, are a small-scaled version of data centers to extend the hyperspace cloud data centers. Different MDCs are connected by the backbone network to achieve more efficient and intelligent computation, caching, and management. MDC also serves as an important role in managing numerous Internet of Things (IoT) devices [21].

2) FOG COMPUTING

Fog computing [3], first proposed by Cisco, is a computing paradigm that aims to bring cloud computing services to the end of the enterprise network. In fog computing, the data processing is carried out at fog nodes, which are usually deployed at the network gateway. The fog computing presents a high-level platform that the numerous IoT devices can be interconnected through the distributed fog nodes to provide collaborative services [22]. The fog nodes are also mainly designed to provide better support for the IoT devices. From this perspective, compared to other similar edge computing paradigms, fog computing often stands in alignment with IoT and emphasizes more on the end side.

3) MOBILE EDGE COMPUTING

The paradigm of mobile edge computing was first standardized by European Telecommunications Standards Institute (ETSI) [23], which aims to provide sufficient computing capacities within the radio access network (RAN). It envisions that the computing capacities are placed at the end of the cellular network, e.g., the wireless base stations. Since base stations are the important access gate for numerous IoT devices, mobile edge computing could provide direct service to the end devices through only one hop, bringing great convenience for IoT data processing [16].

B. ADVANTAGES OF EDGE COMPUTING

Compared to traditional cloud computing, edge computing has many unique advantages, including low latency, energy saving, context-aware service, and privacy as well as security. We next summarize them as follows.

1) LOW LATENCY

Since edge devices are placed closer to end devices, which are usually both the data source and the transmission target of processing results, the transmission latency can be largely reduced compared to the cloud computing scenario. For example, the transmission latency is usually tens (or hundreds) of milliseconds between an end user and a cloud server, while this number is usually several milliseconds or even at microsecond level. The emerging 5G technology further enhances the advances of edge computing from the perspective of low latency transmission, which empowers a series of emerging applications, such as autonomous driving [24], virtual reality/augmented reality and healthcare-related applications.

2) ENERGY SAVING

Restricted by the size and usage scenarios, IoT devices usually have quite limited energy supply, but they are also expected to perform very complex tasks that are usually power consuming. It is challenging to design a cost-efficient solution to well power the numerous distributed IoT devices given that frequent battery charging/discharging is impractical in not possible [16]. Edge computing enables the billions of IoT devices to offload the most power-consuming computation tasks to the edge servers, which not only greatly reduce the power consumption but also improves the processing efficiency.

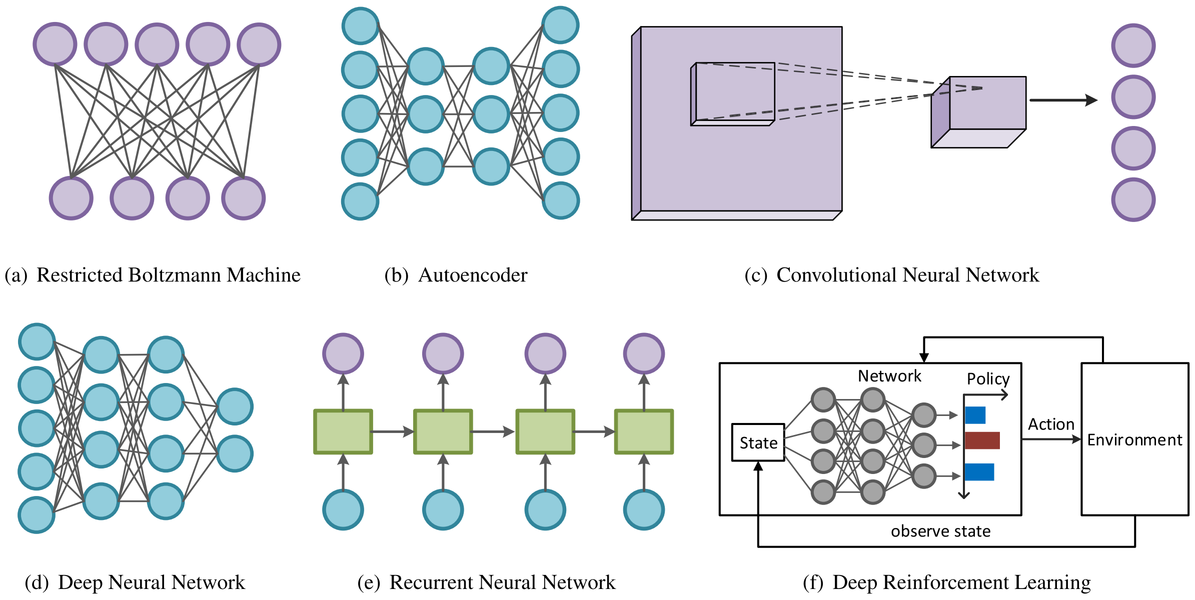
(a) Restricted Boltzmann Machine
(b) Autoencoder
(c) Convolutional Neural Network
(d) Deep Neural Network
(e) Recurrent Neural Network
(f) Deep Reinforcement Learning

FIGURE 2. **The structures of different deep learning models.**

3) CONTEXT-AWARE SERVICE

Context-aware computing [25] is playing an important role in IoT and edge computing applications, since good modeling and reasoning of collected data can highly rely on the context of the data. With the advantage of the proximity nature, edge servers can collect more context information to support the data processing. For example, in the Amazon Go supermarket, video cameras can not only record the goods that customers select but also predict customers' interest based on their staying location, duration and behaviors.

4) PRIVACY AND SECURITY

Compared to cloud computing, edge computing is more efficient and effective in protecting the data privacy and application security of users. On one hand, edge servers are usually geographically distributed clusters that could be managed and maintained by users themselves. Sensitive information can be monitored and protected more strictly. On the other hand, the small-scale nature makes it more concealed than large-scale data centers, further making it less likely to become a target of attacks [26].

III. DEEP LEARNING METHODS

Deep learning has been widely applied in many fields with great success [27], such as computer vision (CV), natural language processing (NLP), and artificial intelligence (AI). Compared to traditional machine learning methods, deep learning has demonstrated powerful information extraction and processing capabilities, but also requires massive computation resources. The breakthroughs of deep learning have greatly expanded the edge computing applications in various scenarios, improving performance, efficiency, and management. In this section, we introduce some typical deep learning models that are widely used for edge computing applications, including restricted Boltzmann machine (RBM), autoencoder (AE), deep neural network (DNN), convolutional neural network (CNN), recurrent neural network (RNN), and deep reinforcement learning (DRL). The basic architectures of these learning models are illustrated in Fig. 2.

A. RESTRICTED BOLTZMANN MACHINE

Restricted Boltzmann machine (RBM) is a kind of probabilistic graphical models that can be interpreted as stochastic neural networks [28]. A typical two-layer RBM includes a visible layer that contains the input we know and a hidden layer that contains the latent variables, as described in Fig. 2(a). RBMs are organized as a bipartite graph, where each visible neuron is connected to all hidden neurons and vice versa, but any two units are not connected in the same layer. RBMs have seen successful applications in many fields, such as collaborative filtering [29] and network anomaly detection [30]. Multiple stacked RBM layers can form a deep belief network (DBN), which consists of a visible layer and multiple hidden layers. The training of a DBN follows a layer-by-layer method, where each layer is treated as an RBM trained on top of the previously trained layer [31]. Many applications can benefit from the structure of DBNs, such as fault detection classification in industrial environments, threat identification in security alert systems, and emotional feature extraction out of images [17].

B. AUTOENCODER

An autoencoder includes an input layer and an output layer that are connected by one or multiple hidden layers [32], as illustrated in Fig. 2(b). The shape of the input layer and the output layer are the same. The AE can be divided into two parts, i.e., an encoder and a decoder. The encoder learns the representative characteristics of the input and transforms it into other latent features (usually in a compressing way). And the decoder receives the latent features of the encoder and aims to reconstruct the original form of the input data, minimizing the reconstruction error. Similarly, an AE can be formed as a deep architecture by stacking multiple layers into the hidden layer. There are several variants and extensions of AEs, such as sparse AE [33], denoising AE [34], and variational AE [35].

C. DEEP NEURAL NETWORKS

Compared to the traditional artificial neural network (ANN) that has shallow structure, deep neural network (DNN) (or deep fully connected neural network) usually has a deeper layer structure for more complicated learning tasks [32]. A DNN consists of an input layer, several hidden layers, and an output layer, where the output of each layer is fed to the next layer with activation functions. At the last layer, the final output representing the model prediction is produced. Optimization algorithms such as Stochastic Gradient Decent (SGD) [36] and backpropagation [37] are mostly used in the training process. DNNs are widely used in feature extraction, classification and function approximation.

D. CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks (CNNs) are designed to process data that comes in the form of multiple arrays, for example, a color image composed of three 2D arrays containing pixel intensities in the three color channels [27]. A CNN receives 2D data structures and extracts high-level features through convolutional layers as described in Fig. 2(c), which is the core of CNN architecture. By going through the 2D data with a set of moving filters and the pooling functions, CNN extracts the spatial correlations between adjacent data by calculating the inner product of the input and the filter. After that, a pooling block is operated over the output to reduce the spatial dimensions and generate a high-level abstraction. Compared to traditional fully connected deep networks, CNN can effectively decrease the parameter numbers of network and extract the spatial correlations in the raw data, mitigating the risk of overfitting [38]. The above advantages make CNN achieve significant results in many applications, such as object detection [39] and health monitoring [40].

E. RECURRENT NEURAL NETWORKS

Different from CNNs that are good at abstracting spatial features, recurrent neural networks (RNNs) are designed for processing sequential or time-series data. The input to an RNN includes both the current sample and the previously observed samples. Specifically, each neuron of an RNN layer not only receives the output of its previous layer but also receives the stored state of from previous time steps, as depicted in Fig. 2(e). With this special architecture, RNN is able to remember previous information for integrated processing with the current information. However, RNNs can only look back for a few steps due to the gradient explosion and long-term dependencies. To solve this problem, Long Short-Term Memory (LSTM) network [41] is proposed to control the flow of information. In LSTM model, the forget gate is utilized to control the cell state and decide what to keep in the memory. Through the learning process, stored computations in the memory cells are not distorted over time, which particularly achieves better performance when data is characterized in long dependency [42]. RNN and LSTM are widely used in various sequential scenarios, such as language processing [43] and activity recognition [44].

F. DEEP REINFORCEMENT LEARNING

Deep reinforcement learning (DRL) [7] is a combination of deep learning (DL) and reinforcement learning (RL) [45]. It aims to build an agent that is able to learn the best action choices over a set of states through the interaction with the environment, so as to maximize the long-term accumulated rewards. Different from traditional RL, DRL utilizes a deep neural network to represent the policy given its strong representation ability to approximate the value function or the direct strategy. DRL can be categorized into value-based models, such as Deep Q-Learning (DQL), Double DQL [46] and Duel DQL [47], and policy-gradient-based models, such as deep deterministic policy gradient (DDPG) [48] and asynchronous advantage actor-critic (A3C) [49]. The DRL has been successfully applied in many fields, such as computer gaming [7], chess gaming [8] and rate adaptation [50].

IV. EMPOWERING EDGE APPLICATIONS WITH DEEP LEARNING

A. WHEN EDGE COMPUTING MEETS DEEP LEARNING

Recent years have witnessed the rapid development and the achieved great success of edge computing and deep learning in their respective fields. However, the massive amount of invaluable data generated and collected at the edge side calls for more powerful and intelligent processing capacities locally to fully unleash the underlying potentials of big data, so as to satisfy the ever-increasing demands of various applications. Fortunately, the recent breakthroughs in deep learning shed a light on the edge application scenarios, providing strong ability in information perception, data management, decision making, etc. The convergence of these two technologies can further create new opportunities, empowering the development of many emerging applications. In fact, edge computing has already been gradually integrated with artificial intelligence (AI) to achieve *edge intelligence*. In this rest of this section, we conduct a comprehensive overview of state-of-the-art research works on edge computing applications with deep learning and summarize them in several aspects, including smart multimedia, smart transportation, smart city, and smart industry.

B. SMART MULTIMEDIA

The Internet video content has been explosively increasing in the past years. As estimated by Cisco, the global video

TABLE 1. Summary of deep-learning-enabled edge computing applications in the field of smart multimedia.

Application	Work	Model	Summary
Video Analytics	Ren et al. [51]	R-CNN	An objection detection architecture and implementation that leverages Faster R-CNN for objection detection.
	Liu et al. [52]	CNN	A CNN-based visual food recognition algorithms with edge computing.
	DeepDecision [53]	CNN, Yolo [54]	A distributed infrastructure that ties end and edges to improve Yolo with higher frame rate and accuracy.
	DeepCham [55]	CNN	Use edge and end user to collaboratively train CNN model for better object recognition accuracy.
	Nikouei et al. [56]	L-CNN	Develop L-CNN for resource-constrained edge devices with reduced filter numbers.
Adaptive Streaming	Grazia [57]	RBM, Liner Classifier	A multi-stage learning system integrating RMB and liner classifier for simultaneous video transmission with guarantees for each user.
	Wang et al. [58]	DRL	An edge computing-assisted framework that leverages DRL to intelligently assign user to proper edge servers.
Caching	Li et al. [59]	ARIMA, MLR,kNN	Consider the video propagation and popularity evolution patterns with ARIMA, MLP, and kNN.
	Zhang et al. [60]	LSTM-C	Propose a caching framework LSTM-C to better learn to content popularity patterns both at long and short time scale.
	Zhu et al. [61]	DRL	Leverage DRL to automatically learn an end-to-end caching policy.
	Jiang et al. [62]	MADRL	Formulate the D2D caching problem as a multi-agent MAB problem and developing a multi-agent DRL solution.

traffic accounted for 75% of the Internet traffic in 2017 and is estimated to grow four-fold by 2022 [63]. Meanwhile, people are having an increasingly higher demand for video content and video watching experience, calling for more intelligent video processing, caching, and delivery, etc. Nowadays, deep learning is integrated with edge computing to provide both better video quality of experience (QoE) for viewers and cost-effective functions to service providers. The representative researches on smart multimedia applications are summarized in Tab. 1.

1) VIDEO ANALYTICS

Nowadays, video analytics [9] are becoming more and more widely used in different fields such as camera-based surveillance [64] and augmented reality (AR). With the limited processing capabilities of cameras, traditional video analytics usually heavily rely on cloud computing for content processing, i.e., the video contents are first streamed to the backend cloud servers and the processed results are then delivered to the frontend devices. This processing mode however brings high latency and consumes much bandwidth, unable to satisfy those latency-sensitive applications, not to mention those realtime requirements such as object detection [51] and tracking [65]. The emergence of edge computing pushes the video analytics from the remote cloud to the local edge, allowing the video content to be processed near the data source so as to enable quick or even realtime response. For example, Amazon has released the world's first deep-learning-enabled video camera, where the locally executed deep learning function enables realtime objection even without the involvement of the cloud.

Pioneer researches have conducted efforts towards intelligent video analytics with edge computing. Ren *et al.* [51] proposed an edge-computing-based object detection architecture as well as a preliminary implementation to achieve distributed and efficient object detection via wireless communications for real-time surveillance applications. They adopted Faster R-CNN [39] for model training and object detection, with a well-designed RoI detection algorithm to balance the detection accuracy and the data compression rate. Liu *et al.* [52] developed a CNN-based visual food recognition algorithms to achieve the best-in-class recognition accuracy, where edge computing was employed to overcome the system latency and low battery life of mobile devices. In DeepDecision [53], a distributed infrastructure was proposed to tie together computationally weak frontend devices (assumed to be smartphones) with more powerful back-end helpers (the edges) to allow deep learning to choose local or remote execution. This approach boosts the performance of CNN, in particular Yolo [54], to achieve higher frame rate and accuracy. DeepCham [55] leveraged an edge master server coordinated with several participating users to collaboratively train CNN model to achieve better object recognition accuracy. LAVEA [66] built up a client-edge collaboration system and solved an optimization offloading problem to minimize the response time. Nikouei *et al.* [56] developed a lightweight convolutional neural network (L-CNN), which leveraged the depthwise separable convolution feature and tailored the CNN to be furnished in the resource-constrained edge devices with reduced filter numbers in each layer.

2) ADAPTIVE STREAMING

Adaptive video streaming [78] is becoming a critical issue in today's video delivery to provide the best quality of experience (QoE) over the Internet. The basic idea is to select proper video bitrate considering the network states, stability, fairness, user's preference of video quality, etc. Most existing adaptive streaming approaches [79]–[81] rely on the client-based adaptation, which aims to adapt to the bandwidth variations based on several observed or predicted metrics such as buffer size and bandwidth situation. The recent advances in deep learning, particularly deep reinforcement learning, are leveraged at the client end to automatically learn the adaptation policy for better QoE [50], [82].

The emergence of edge computing provides an alternative for adaptive video streaming. Given the intelligence and computation capability, edge nodes can serve as a cache server or transcoding server to provide edge-based (or edge-cloud-based) adaptive streaming [83], [84]. This scheme can usually collect more states from other users and achieve higher fairness, stability, and collaborative intelligence. For example, De Grazia [57] developed a multi-stage learning system to manage simultaneous video transmission which guarantees a minimum quality level for each user. In particular, they used an unsupervised Restricted Boltzmann Machine (RBM) [28] to capture the latent features of the input data and a supervised linear classifier to estimate the characteristics of unknown videos. Wang *et al.* [58] designed an edge computing-assisted framework that leverages DRL to intelligently assign users to proper edge servers to achieve proper video streaming services.

3) CACHING

Video content caching [85] is another important application that has attracted continuous research efforts for years given its great benefits in improving multimedia services. In traditional content delivery network (CDN) architecture, video contents are usually placed or cached at remote servers, where the high latency and limited bandwidth between the end viewers and the remote servers can cause viewing delay and congestion, seriously undermining the viewers' QoE. The emerging edge caching [86], [87] is able to alleviate this problem by pushing the content close to the end users so as to reduce the access latency and reduce the network pressure. Traditional content providers may simply use rule-based solutions such as Least Recently Used (LRU), Least Frequently Used (LFU) and their variants [88], [89], or model-based solutions such as [90], [91] given the easy implementation. These solutions however heavily rely on dedicated features and are not adaptive enough to the changing characteristics.

Deep learning brings new opportunities towards intelligent edge caching using advanced learning techniques to well capture the hidden features. Li *et al.* [59] considered the video propagation as well as popularity evolution patterns and developed an integration of ARIMA, multiple linear regression (MLR), and k-nearest neighbor regression (kNN) to predict the social patterns to improve caching performance. Zhang *et al.* [60] proposed an intelligent edge-assisted caching framework LSTM-C based on LSTM to better learn to content popularity patterns both at long and short time scale. Zhu *et al.* [61] proposed to leverage DRL to automatically learn an end-to-end caching policy, where the user requests, network constraints, and external information are all embedded in the learning environment. Besides individual caching decisions, collaborative caching is also explored in recent years to achieve collective intelligence. For example, Jiang *et al.* [62] formulates the D2D caching problem as a multi-agent multi-armed bandit (MAB) problem and developed a multi-agent DRL (MADRL) solution to learn a coordinated caching scheme among multiple agents.

C. SMART TRANSPORTATION

Vehicle is envisioned as the next intelligent information carrier after smartphone. The coming era of 5G and mobile edge computing (MEC) has enabled vehicle information to be readily accessible anytime and anywhere with low latency, forming an Internet of Vehicle (IoV) [92]. Integrated with the latest advances in deep learning, IoV will enable more intelligent transportation management, such as autonomous driving [24], traffic prediction, traffic signal control, as summarized in Tab. 2.

1) AUTONOMOUS DRIVING

Intelligent sensing and perception are of the most critical issues in autonomous driving [24]. The vehicles first collect the information from various carried sensors such as cameras and radars, and then conduct an intelligent perception and decision. Purely using vehicle-based and cloud-based solutions may not well satisfy the requirement of high computation capacity, realtime feedback, enough redundancy, and security for autonomous driving. Edge computing however provides a promising solution with powerful computation and low-latency communication [105]. With the benefits of V2X communications [106], part of the learning-based perception can be offloaded to the edge server for processing.

Many existing works have conducted pioneer efforts towards autonomous driving. SqueezeDet [67] proposed a carefully designed CNN-based learning pipeline that not only achieves high object detection accuracy but also reduces the model size for energy efficiency. To better understand the captured object, Chen *et al.* [68] proposed a monocular 3D object detection method using state-of-the-art CNN method based on the fact that objects should be on the ground-plane. To further improve the accuracy and robustness, MV3D [69] developed a multi-view 3D deep learning network that takes both LIDAR point and camera images as a fusion input to predict 3D boundaries. Besides the road feature extraction, researchers also dived deeply into the driving control based on the sensing information. Bojarski *et al.* [70] proposed an end-to-end learning architecture without detecting road features. It directly mapped the raw pixels from a single front-facing camera to the steering commands.

2) TRAFFIC ANALYSIS AND PREDICTION

Understanding the mobility patterns of the vehicles and people is a critical problem for urban traffic management, city planning, and service provisioning. Given the distributed features of mobile edge servers, edge computing is naturally ideal for vehicle traffic analysis and prediction [107]. Traditional approaches mostly used time-series analysis [108] or probabilistic graph analysis [109], which may not sufficiently

TABLE 2. Summary of deep-learning-enabled edge computing applications in the field of smart transportation.

Application	Work	Model	Summary
Autonomous Driving	SqueezeDet [67]	CNN	A careful designed CNN for objection with reduced model size.
	Chen et al. [68]	CNN	A CNN-based monocular 3D object detection method based on the fact that objects should be on the ground-plane.
	MV3D [69]	Fusion VGG	A multi-view 3D deep learning network that takes both LIDAR point and camera images as a fusion input.
	Bojarski et al. [70]	CNN	An end-to-end learning architecture that directly mapped the raw pixels from a single front-facing camera to the steering commands.
Traffic Analysis and Prediction	Polson et al. [71]	DNN	Leveraging deep neural network to mine the short term characteristics of the traffic situation for traffic prediction.
	Lv et al. [72]	SAE	Leverage a stacked autoencoder (SAE) to learn the generic traffic features from the historical data.
	Koesdwiady et al. [73]	DBN	Consider the impact of weather on traffic situation and incorporat the weather information into a deep belief network.
	DeepTransport [74]	LSTM	Consider the mobility analysis at the citywide scale with a LSTM model for prediction.
	Yao et al. [75]	CNN,LSTM	Revisit the spatiotemporal relationships in traffic pattern and propose a novel spatial-temporal dynamic network (STDN) based on CNN and LSTM for prediction.
Traffic Signal Control	Wiering et al. [76]	Tabular QL	Leverage a tabular Q-learning model in an isolated intersection for signal control.
	Abdoos et al. [10]	MAQL	Propose a multi-agent Q-learning (MAQL) method that considers the queue length for cooperative scheduling.
	Chu et al. [77]	MA2C	Propose a novel multi-agent actor-critic (MA2C) approach to comprehensively combine the traffic features for intelligent control.

capture the hidden spatiotemporal relationships therein. As a powerful learning tool, deep learning stands out as an effective method in this direction. Liu *et al.* [110] further pointed out the potential of applying different deep learning approaches in urban traffic prediction. Polson *et al.* [71] leveraged deep neural network to mine the short term characteristics of the traffic situation of a road segment to predict the near future traffic pattern. Lv *et al.* [72] leveraged a stacked autoencoder (SAE) to learn the generic traffic features from the historical data. Koesdwiady *et al.* [73] further considered the impact of weather on traffic situations and incorporated the weather information into a deep belief network for integrated learning. In DeepTransport [74], the authors considered the mobility analysis at a larger scale, i.e., the citywide scale. LSTM model is used for future movement prediction. In [75], the authors revisited the spatiotemporal relationships in traffic patterns and proposed a novel spatial-temporal dynamic network (STDN) based on CNN and LSTM, which outperforms the existing prediction methods.

3) TRAFFIC SIGNAL CONTROL

With the above traffic analysis and prediction, the combination of edge computing and deep learning actually can do more things towards intelligent transportation management. Among them, intelligent traffic signal control [76] is one of the most representative applications and has also been explored by researchers for years. A good control policy is able to reduce the average waiting time, traffic congestion, and traffic accident. The early traffic signal control methods usually rely on fuzzy logic [111] or genetic algorithm [112]. The key challenge however lies in how to achieve collaborative and intelligent control among multiple or even citywide traffic lights for large scale traffic scheduling. Towards this goal, reinforcement learning (RL) and multi-agent RL turns out to be a promising solution where each agent (can be implemented as an edge) will make control policy for a traffic light considering not only its local traffic situation but also other agents' traffic situations. In [76], tabular Q-learning was first applied in an isolated intersection for signal control. To improve the collaboration among traffic lights, Abdoos *et al.* [10] proposed a multi-agent Q-learning (MAQL) method that considered the queue length for cooperative scheduling. The latest work [77] further integrated the state-of-the-art actor-critic (A2C) RL algorithm and the multi-agent learning as a multi-agent actor-critic (MA2C) approach to comprehensively combine the traffic features for intelligent control.

D. SMART CITY

Smart city [113] is another important application scenario for deep-learning-enabled edge computing. The geo-distributed big data [114] in a city naturally requires a distributed computing paradigm for local processing and management. The integration of edge computing and deep learning enables the deep penetration of computing intelligence into every corner of a city, forming a smart city that can provide more efficient, economic, energy-saving, and convenient services [115], [116]. We next discuss the combinational advantages in the main components of the smart city, including the smart home, smart building, and smart grid, as in Tab. 3.

1) SMART HOME

Smart IoT has been widely explored in smart home scenarios to provide not only convenient control but also intelligent sensing [117]. Considering the privacy issue in

TABLE 3. Summary of deep-learning-enabled edge computing applications in the field of smart city.

Application	Work	Model	Summary
Smart Home	Dhakal et al. [93]	kNN,DNN	Develop an automated home/business monitoring system on NFV edge servers performing online learning on streaming data from home.
	SignFi [94]	CNN	Exploit the CSI signals of WiFi and using CNN to identify 276 sign language gestures.
	Wang et al. [95]	CNN,LSTM	Leverage a combined CNN and LSTM to recognize different gestures and activities.
	Mohammadi et al. [96]	DRL	Propose a novel semisupervised DRL-based method for indoor localization.
Smart Building	Zheng et al. [97], [98]	Multi-task Learning	Leverage multi-task learning to predict the performance of a chiller and strike balance between electricity consumption and real-world deployment.
	Yuce et al. [99]	NN	Propose a neural network based model to perform regression analysis of energy consumption within a building.
	Thokala [100]	SVM,RNN	Consider the heterogeneity in the electrical load and propose to use both SVM and partial RNN to forecast future load.
Smart Grid	He et al. [101]	DBN,RBM	A deep learning based mechanism that integrates DBN and RBM to detect the attack behavior of false data injection in realtime.
	Yan et al. [102]	RL	A reinforcement learning-based approach to identify critical attack sequences with consideration of physical system behaviors.
	Shi et al. [103]	RNN	A novel pooling-based RNN network to forecast the household load addressing the over-fitting issue.
	Wan et al. [104]	DRL	A model-free DRL-based model to automatically determine the charging policy.

the home scenario, edge computing is a good choice to provide local computation and processing, especially for the computation-intensive deep-learning-based applications. Dhakal AND Ramakrishnan [93] developed an automated home/business monitoring system which resides on Network Function Virtualization (NFV) edge servers performing online learning on streaming data coming from homes and businesses in the neighborhood. Leveraging the latest advances of deep learning, the ubiquitous wireless signals can also be used for smart interaction between human and devices. For example, SignFi [94] exploited the CSI signals of WiFi and was able to identify 276 sign language gestures including the head, arm, hand, and finger with CNN for classification. Wang *et al.* [95] analyzed the impact patterns of moving humans on the WiFi signals and leveraged a combined CNN and LSTM to recognize different gestures and activities. Such method can be used for remote control of home devices such as lights and televisions [118]. Mohammadi *et al.* [96] explored more possibilities of deep learning approaches and proposed a novel semisupervised DRL based method for indoor localization. Such edge-intelligence-enabled solution can be widely used for smart home, including intrusion detection, gesture-based interaction, fall detection, etc.

2) SMART BUILDING

Achieving intelligent monitoring, sensing, and control in the building environment requires more comprehensive perception and processing compared to the home environment given the complex architecture. In this context, edge computing plays an important role in data processing, orchestration, and privacy preserving. Recently, many efforts [119]–[121] have been made towards smart building to reduce energy consumption, improve the building security, enhance the sensing capacity of buildings, etc. Zheng *et al.* [97], [98] focused on the chiller sequencing problem to reduce the electricity consumption in buildings. They leveraged multi-task learning to predict the performance of a chiller and further strike a good balance between the electricity consumption and ease of use for real-world deployment. Yuce and Rezgui [99] proposed a neural-network-based model to perform regression analysis of energy consumption within a building. Thokala [100] further considered the heterogeneity in the electrical load and proposed to use both SVM and partial RNN to forecast future load.

3) SMART GRID

The smart grid is an electricity distribution network with smart meters deployed at various locations to measure the realtime status information [127]. Smart grid is also an important use case for edge computing or fog computing. Edge collectors at the edge ingest the data generated by grid sensors and devices, where some data for protection and control loops even require real-time processing (from milliseconds to sub-seconds) [3]. Deep learning together with the edge computing empowers the grid with more intelligent protection, control, and management. He *et al.* [101] proposed a deep-learning-based mechanism that integrated deep belief network (DBN) and RBM to detect the attack behavior of false data injection in realtime. Considering the sequential behavior of attacks for smart grid, Yan *et al.* [102] further proposed a reinforcement learning-based approach to identify critical attack sequences with consideration of physical system behaviors. Shi *et al.* [103] proposed a novel pooling-based RNN network to forecast the household load addressing the over-fitting issue. Pricing is another important issue towards smart grid, which greatly affects customers' using behaviors in many aspects, e.g., the economy-driven

TABLE 4. Summary of deep-learning-enabled edge computing applications in the field of smart industry.

Application	Work	Model	Summary
Smart Manufacturing	Weimer et al. [122]	CNN	A novel CNN-based architecture for fast and reliable industrial inspection.
	Li et al. [123]	CNN	An edge-computing-based model that offloads computation burden to the fog nodes and a CNN-based model with an early-exit design for accurate inspection.
	Zhao et al. [124]	CNN, bi-LSTM	A system that combined CNN and bi-directional LSTM for machine health monitoring.
Smart Industrial Analysis	Wu et al. [125]	vanilla LSTM	A remaining life prediction for engineered system using vanilla LSTM neural networks.
	Wang et al. [126]	DNN	Proposing a DNN-based architecture to accurately predict the remaining energy and remaining lifetime of batteries, which further enables an informed power configuration among base stations.

electric vehicle charging [128]. For instance, Wan *et al.* [104] jointly considered the electricity price and battery energy of electric vehicles and proposed a model-free DRL based model to automatically determine the charging policy.

E. SMART INDUSTRY

In the coming era of industry 4.0, we are experiencing a revolution of smart industry, which has two main principles, i.e., production automation and smart data analysis [123]. The former is one of our main objectives that could greatly liberate the productivity and the latter is one of the most effective methods towards our objectives. The recent advances of edge computing migrate the massive computation from the remote cloud to the local edge, enabling more low-latency and secure manufacturing [129]. And deep learning further empowers more effective local analysis and prediction [130] at the edge node of industry instead of the cloud. We summarize them in the next two parts as illustrated in Tab. 4.

1) SMART MANUFACTURING

Smart manufacturing is the key component of the smart industry, which highly relies on the intelligent processing of deep learning and quick response of edge computing. The combination of deep learning and edge computing has been applied in many aspects of industry manufacturing, such as manufacture inspection and fault assessment. Weimer *et al.* [122] developed a novel CNN-based architecture for fast and reliable industrial inspection, which can automatically generate meaningful features for a specific inspection task from a huge amount of raw data with minimal human interaction. Li *et al.* [123] proposed an edge-computing-based model that is able to offload the computation burden to the fog nodes to deal with extremely large data. A CNN-based model together with an early-exit design is used in this model, which largely improved the inspection accuracy and robustness. Zhao *et al.* [124] further combined CNN with bi-directional LSTM to propose a novel machine health monitoring system.

2) SMART INDUSTRIAL ANALYSIS

Besides the manufacture inspection and monitoring, the application of edge computing and deep learning also enables much intelligent industrial analysis. Wu *et al.* [125] focused on the remaining useful life estimation of the engineered system and proposed to use vanilla LSTM neural networks to get good remaining lifetime prediction accuracy. Wang *et al.* [126] focused on the remaining lifetime analysis of backup batteries in the wireless base stations. They proposed to use DNN-based architecture to accurately predict the remaining energy and remaining lifetime of batteries, which further enables an informed power configuration among base stations.

V. RESEARCH CHALLENGES AND DIRECTIONS

Though the convergence of edge computing and deep learning has revealed great potentials and prompted the fast development of many applications, there still exist various problems in achieving stable, robust, and practical usage, which calls for continuous efforts in this field from many perspectives. We next discuss some key research challenges and promising directions.

A. MODEL TRAINING

The performance of deep-learning-enabled edge applications highly relies on how the learning models are performed in the edge computing architecture, where the model training is an important process. It is well known that model training is often computation-intensive, consuming massive CPU and GPU resources, especially for those deep models. Edge servers are usually challenging, or at least not cost-efficient, to solely take the model training tasks. Besides, in many applications, the data is distributed collected from multiple edge servers and it is difficult for a single edge server to obtain the whole information for model training. Sharing the raw data among the edge nodes is not a good solution since it will consume massive communication resources. Towards this direction, distributed learning [131] and federated learning [132] are two promising models to address this problem. The early idea of distributed learning is to design a decentralized Stochastic Gradient Descent (SGD) algorithm in the edge computing environment. The key challenge exists in reducing the communication cost for gradient updates while preserving the training accuracy. Recent efforts have been made towards this direction, e.g., delivering the important gradient first [133]. Federated learning is another promising method emerging in recent years to train deep neural networks as it leaves the raw data on clients and only aggregates the intermediate updates from each client. In the edge application scenario, it can also reduce the communication cost and improve resource utilization [134].

B. MODEL INFERENCE

As many outstanding learning models are getting bigger and deeper, traditional large-scale learning models are often deployed in a centralized cloud and receive the raw input data from the distributed end devices, which can cause high delay. Edge servers provide alternative solutions for model deployment, where the edge and cloud can work collaboratively to handle the massive amounts of learning takes. A promising direction is model partition for deep neural networks, where the end, edge, and cloud will execute part of the learning models, respectively. For example, Kang *et al.* [135] developed a lightweight scheduler to automatically partition DNN computation between mobile devices and data centers at the granularity of neural network layers. And Huang *et al.* [136] further explored the partitioning problem in the edge computing scenario, and developed a partitioning strategy among the device, edge, and cloud, which aimed to reduce the execution delay. Another promising direction is the early exit of inference (EEoI) [137] for deep neural networks. Since passing through the whole deep networks is both time- and energy-consuming for edge servers, EEoI allows the inference to exit early if verified by some predefined models.

C. APPLICATION ENHANCEMENT

The integration of deep learning and edge computing has achieved remarkable improvement for many application scenarios, yet there are still some critical applications desiring for real breakthroughs. Real-time VR gaming and autonomous driving are two most representative applications. Both these two applications require ultra low-delay interactions and powerful computations. The emerging 5G communication technology together with edge learning brings new possibilities towards a feasible solution, where the main computation such as video rendering as well as video analytics can be conducted at local edges and the processing results can be delivered to the end in near real time, e.g., millisecond-level interaction. Despite of the preliminary foundation of feasibility, there is still a long way to go before the practical application.

D. HARDWARE AND SOFTWARE OPTIMIZATION

In addition to the model and application enhancement, the system-level optimization for deep learning and edge computing is also a challenging yet promising direction. Most of the existing hardware architecture, software platform, and programming abstraction are particularly designed for the cloud-based computing paradigm. Yet the edge learning emphasizes some different aspects compared to cloud computing, such as energy-efficiency, light-weight architecture, and edge-oriented computation framework. For example, from the perspective of the hardware architecture optimization, Du *et al.* [138] studied Cortex-M micro-controllers and proposed a streaming hardware accelerator to better CNN in edge devices. Besides, FPGA-based edge computing platforms are also developed to support deep learning computation offloading from mobile devices to the edge FPGA platform [139]. For the software perspective, many incorporations have proposed their own software platforms or services to support edge-level learning and computing, such as Amazon's Greengrass[2] and Microsoft's Azure IoT Edge.[3] For the perspective of programming abstraction, there are also some frameworks specially designed for edge scenario, such as MXNet [140], Tensorflow Lite[4] and CoreML.[5] Though with these existing systems and frameworks, there still need a lot of efforts to integrate them to achieve a more practical and high-performance system for general edge learning applications.

VI. CONCLUSION

In this article, we mainly investigated how the recent advances of deep learning can be leveraged to improve the novel edge computing applications. We first introduced the basic concepts and paradigms of edge computing, highlighting its key advantages. We then present some representative deep learning models that can be used in edge computing, such as autoencoder, CNN, RNN, DRL, etc. A comprehensive survey on the latest deep learning empowered edge computing applications is next conducted from four domains, including smart multimedia, smart transportation, smart city, and smart industry. Finally, we discussed the key challenges and future research directions on improving the intelligent edge computing applications. We hope this survey is able to elicit more discussion and inspiration on the convergence of edge computing and deep learning, so as to facilitate the development of deep-learning-enabled edge computing applications.

REFERENCES

[1] *Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are, White Paper*. Accessed: Mar. 21, 2020. [Online]. Available: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. ACM MCC Workshop*, 2012, pp. 13–16.

[4] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervas. Comput.*, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NeurIPS*, 2012, pp. 1097–1105.

[6] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.

[2] https://aws.amazon.com/cn/greengrass/

[3] https://azure.microsoft.com/en-ca/services/iot-edge/

[4] https://www.tensorflow.org/lite

[5] https://developer.apple.com/documentation/coreml?language=objc

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, ''Human-level control through deep reinforcement learning,'' *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[8] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, ''Mastering the game of go with deep neural networks and tree search,'' *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[9] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, ''Real-time video analytics: The killer app for edge computing,'' *Computer*, vol. 50, no. 10, pp. 58–67, 2017.

[10] M. Abdoos, N. Mozayani, and A. L. C. Bazzan, ''Traffic light control in non-stationary environments based on multi agent Q-learning,'' in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 1580–1585.

[11] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, ''Convergence of edge computing and deep learning: A comprehensive survey,'' 2019, *arXiv:1907.08349*. [Online]. Available: http://arxiv.org/abs/1907.08349

[12] J. Chen and X. Ran, ''Deep learning with edge computing: A review,'' *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.

[13] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, ''Edge intelligence: Paving the last mile of artificial intelligence with edge computing,'' 2019, *arXiv:1905.10083*. [Online]. Available: http://arxiv.org/abs/1905.10083

[14] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, ''A survey on mobile edge networks: Convergence of computing, caching and communications,'' *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[15] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, ''Mobile edge computing: A survey,'' *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[16] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, ''A survey on mobile edge computing: The communication perspective,'' *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[17] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, ''Deep learning for IoT big data and streaming analytics: A survey,'' *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2923–2960, 4th Quart., 2018.

[18] X. Ma, T. Yao, M. Hu, Y. Dong, W. Liu, F. Wang, and J. Liu, ''A survey on deep learning empowered IoT applications,'' *IEEE Access*, vol. 7, pp. 181721–181732, 2019.

[19] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, ''Cloudlets: Bringing the cloud to the mobile user,'' in *Proc. ACM MCS*, 2012, pp. 29–36.

[20] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, ''Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers,'' *Comput. Netw.*, vol. 130, pp. 94–120, Jan. 2018.

[21] M. Aazam and E.-N. Huh, ''Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT,'' in *Proc. IEEE AINA*, Mar. 2015, pp. 687–694.

[22] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, ''Fog computing: A platform for Internet of Things and analytics,'' in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Cham, Switzerland: Springer, 2014, pp. 169–186.

[23] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, ''Mobile edge computing—A key technology towards 5G,'' *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.

[24] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, ''Edge computing for autonomous driving: Opportunities and challenges,'' *Proc. IEEE*, vol. 107, no. 8, pp. 1697–1716, Aug. 2019.

[25] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, ''Context aware computing for the Internet of Things: A survey,'' *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414–454, 1st Quart., 2014.

[26] C. Wang, G. Liu, H. Huang, W. Feng, K. Peng, and L. Wang, ''MIASec: Enabling data indistinguishability against membership inference attacks in MLaaS,'' *IEEE Trans. Sustain. Comput.*, early access, Jul. 23, 2019, doi: 10.1109/TSUSC.2019.2930526.

[27] Y. LeCun, Y. Bengio, and G. Hinton, ''Deep learning,'' *Nature*, vol. 521, no. 7553, p. 436, 2015.

[28] A. Fischer and C. Igel, ''An introduction to restricted Boltzmann machines,'' in *Proc. Iberoamer. Congr. Pattern Recognit.* Berlin, Germany: Springer, 2012, pp. 14–36.

[29] R. Salakhutdinov, A. Mnih, and G. Hinton, ''Restricted Boltzmann machines for collaborative filtering,'' in *Proc. ACM ICML*, 2007, pp. 791–798.

[30] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, ''Network anomaly detection with the restricted Boltzmann machine,'' *Neurocomputing*, vol. 122, pp. 13–23, Dec. 2013.

[31] G. E. Hinton, S. Osindero, and Y.-W. Teh, ''A fast learning algorithm for deep belief nets,'' *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[33] H. Lee, A. Battle, R. Raina, and A. Y. Ng, ''Efficient sparse coding algorithms,'' in *Proc. NeurIPS*, 2007, pp. 801–808.

[34] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, ''Extracting and composing robust features with denoising autoencoders,'' in *Proc. ACM ICML*, 2008, pp. 1096–1103.

[35] C. Doersch, ''Tutorial on variational autoencoders,'' 2016, *arXiv:1606.05908*. [Online]. Available: http://arxiv.org/abs/1606.05908

[36] L. Bottou, ''Large-scale machine learning with stochastic gradient descent,'' in *Proc. COMPSTAT*. Springer, 2010, pp. 17–186.

[37] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, ''Learning representations by back-propagating errors,'' *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[38] M. D. Zeiler and R. Fergus, ''Visualizing and understanding convolutional networks,'' in *Proc. ECCV*. Cham, Switzerland: Springer, 2014, pp. 818–833.

[39] S. Ren, K. He, R. Girshick, and J. Sun, ''Faster R-CNN: Towards real-time object detection with region proposal networks,'' in *Proc. NeurIPS*, 2015, pp. 91–99.

[40] N. Y. Hammerla, S. Halloran, and T. Ploetz, ''Deep, convolutional, and recurrent models for human activity recognition using wearables,'' 2016, *arXiv:1604.08880*. [Online]. Available: http://arxiv.org/abs/1604.08880

[41] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, ''Empirical evaluation of gated recurrent neural networks on sequence modeling,'' 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555

[43] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen, ''Enhanced LSTM for natural language inference,'' 2016, *arXiv:1609.06038*. [Online]. Available: http://arxiv.org/abs/1609.06038

[44] Y. Guan and T. Plötz, ''Ensembles of deep LSTM learners for activity recognition using wearables,'' *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 2, pp. 1–28, Jun. 2017.

[45] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, vol. 2, no. 4. Cambridge, MA, USA: MIT Press, 1998.

[46] H. Van Hasselt, A. Guez, and D. Silver, ''Deep reinforcement learning with double Q-learning,'' in *Proc. AAAI*, 2016, pp. 2094–2100.

[47] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, ''Dueling network architectures for deep reinforcement learning,'' 2015, *arXiv:1511.06581*. [Online]. Available: http://arxiv.org/abs/1511.06581

[48] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, ''Continuous control with deep reinforcement learning,'' 2015, *arXiv:1509.02971*. [Online]. Available: http://arxiv.org/abs/1509.02971

[49] V. Mnih and A. P. Badia, ''Asynchronous methods for deep reinforcement learning,'' in *Proc. ICML*, 2016, pp. 1928–1937.

[50] H. Mao, R. Netravali, and M. Alizadeh, ''Neural adaptive video streaming with pensieve,'' in *Proc. ACM SIGCOMM*, 2017, pp. 197–210.

[51] J. Ren, Y. Guo, D. Zhang, Q. Liu, and Y. Zhang, ''Distributed and efficient object detection in edge computing: Challenges and solutions,'' *IEEE Netw.*, vol. 32, no. 6, pp. 137–143, Nov. 2018.

[52] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, M. Yunsheng, S. Chen, and P. Hou, ''A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure,'' *IEEE Trans. Services Comput.*, vol. 11, no. 2, pp. 249–261, Mar. 2018.

[53] X. Ran, H. Chen, X. Zhu, Z. Liu, and J. Chen, ''DeepDecision: A mobile deep learning framework for edge video analytics,'' in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 1421–1429.

[54] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE CVPR*, Jul. 2017, pp. 7263–7271.

[55] D. Li, T. Salonidis, N. V. Desai, and M. C. Chuah, "DeepCham: Collaborative edge-mediated adaptive deep learning for mobile object recognition," in *Proc. IEEE/ACM SEC*, Oct. 2016, pp. 64–76.

[56] S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B.-Y. Choi, and T. Faughnan, "Smart surveillance as an edge network service: From harr-cascade, SVM to a lightweight CNN," in *Proc. IEEE CIC*, Oct. 2018, pp. 256–265.

[57] M. De Filippo De Grazia, D. Zucchetto, A. Testolin, A. Zanella, M. Zorzi, and M. Zorzi, "QoE multi-stage machine learning for dynamic video streaming," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 1, pp. 146–161, Mar. 2018.

[58] F. Wang, C. Zhang, F. Wang, J. Liu, Y. Zhu, H. Pang, and L. Sun, "Intelligent edge-assisted crowdcast with deep reinforcement learning for personalized QoE," in *Proc. IEEE INFOCOM*, Apr. 2019, pp. 910–918.

[59] H. Li, X. Ma, F. Wang, J. Liu, and K. Xu, "On popularity prediction of videos shared in online social networks," in *Proc. ACM CIKM*, 2013, pp. 169–178.

[60] C. Zhang, H. Pang, J. Liu, S. Tang, R. Zhang, D. Wang, and L. Sun, "Toward edge-assisted video content intelligent caching with long short-term memory learning," *IEEE Access*, vol. 7, pp. 152832–152846, 2019.

[61] H. Zhu, Y. Cao, W. Wang, T. Jiang, and S. Jin, "Deep reinforcement learning for mobile edge caching: Review, new features, and open issues," *IEEE Netw.*, vol. 32, no. 6, pp. 50–57, Nov. 2018.

[62] W. Jiang, G. Feng, S. Qin, T. S. P. Yum, and G. Cao, "Multi-agent reinforcement learning for efficient content caching in mobile D2D networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1610–1622, Mar. 2019.

[63] *Cisco Visual Networking Index: Forecast and Trends, 2017 to 2022 White Paper*. Accessed: Mar. 21, 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html

[64] P. Natarajan, P. K. Atrey, and M. Kankanhalli, "Multi-camera coordination and control in surveillance systems: A survey," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 11, no. 4, p. 57, 2015.

[65] C.-C. Hung, G. Ananthanarayanan, P. Bodik, L. Golubchik, M. Yu, P. Bahl, and M. Philipose, "VideoEdge: Processing camera streams using hierarchical clusters," in *Proc. IEEE/ACM SEC*, Oct. 2018, pp. 115–131.

[66] S. Yi, Z. Hao, Q. Zhang, Q. Zhang, W. Shi, and Q. Li, "LAVEA: Latency-aware video analytics on edge computing platform," in *Proc. ACM/IEEE SEC*, Jun. 2017, p. 15.

[67] B. Wu, A. Wan, F. Iandola, P. H. Jin, and K. Keutzer, "SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *Proc. IEEE CVPR Workshops*, Jul. 2017, pp. 129–137.

[68] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *Proc. IEEE CVPR*, Jun. 2016, pp. 2147–2156.

[69] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE CVPR*, Jul. 2017, pp. 1907–1915.

[70] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*. [Online]. Available: http://arxiv.org/abs/1604.07316

[71] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 1–17, Jun. 2017.

[72] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Sep. 2014.

[73] A. Koesdwiady, R. Soua, and F. Karray, "Improving traffic flow prediction with weather information in connected cars: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9508–9517, Dec. 2016.

[74] X. Song, H. Kanasugi, and R. Shibasaki, "DeepTransport: Prediction and simulation of human mobility and transportation mode at a citywide level," in *Proc. IJCAI*, vol. 16, 2016, pp. 2618–2624.

[75] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 5668–5675.

[76] M. Wiering, J. V. Veenen, J. Vreeken, and A. Koopman, "Intelligent traffic light control," Inst. Inf. Comput. Sci., Utrecht Univ., Utrecht, The Netherlands, Tech. Rep. UU-CS-2004-029, 2004.

[77] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086–1095, Mar. 2020.

[78] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A survey on bitrate adaptation schemes for streaming media over HTTP," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 562–585, 1st Quart., 2019.

[79] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[80] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli, "CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction," in *Proc. ACM SIGCOMM*, 2016, pp. 272–285.

[81] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, 2014, pp. 187–198.

[82] T. Huang, R.-X. Zhang, C. Zhou, and L. Sun, "QARC: Video quality aware rate control for real-time video streaming based on deep reinforcement learning," in *Proc. ACM Multimedia*, 2018, pp. 1208–1216.

[83] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[84] D. Wang, Y. Peng, X. Ma, W. Ding, H. Jiang, F. Chen, and J. Liu, "Adaptive wireless video streaming based on edge computing: Opportunities and approaches," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 685–697, Sep. 2019.

[85] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[86] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[87] E. Zeydan, E. Bastug, M. Bennis, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, "Big data caching for networking: Moving from cloud to edge," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 36–42, Sep. 2016.

[88] S. Podlipnig and L. Böszörmenyi, "A survey of Web cache replacement strategies," *ACM Comput. Surveys*, vol. 35, no. 4, pp. 374–398, Dec. 2003.

[89] Q. Huang, K. Birman, R. van Renesse, W. Lloyd, S. Kumar, and H. C. Li, "An analysis of Facebook photo caching," in *Proc. ACM SOSP*, 2013, pp. 167–181.

[90] J. Hachem, N. Karamchandani, and S. Diggavi, "Content caching and delivery over heterogeneous wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2015, pp. 756–764.

[91] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[92] F. Wang, F. Wang, X. Ma, and J. Liu, "Demystifying the crowd intelligence in last mile parcel delivery for smart cities," *IEEE Netw.*, vol. 33, no. 2, pp. 23–29, Mar. 2019.

[93] A. Dhakal and K. K. Ramakrishnan, "Machine learning at the network edge for automated home intrusion monitoring," in *Proc. IEEE ICNP*, Oct. 2017, pp. 1–6.

[94] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "SignFi: Sign language recognition using WiFi," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–21, Mar. 2018.

[95] F. Wang, W. Gong, and J. Liu, "On spatial diversity in WiFi-based human activity recognition: A deep learning-based approach," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2035–2047, Apr. 2019.

[96] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh, "Semisupervised deep reinforcement learning in support of IoT and smart city services," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 624–635, Apr. 2018.

[97] Z. Zheng, Q. Chen, C. Fan, N. Guan, A. Vishwanath, D. Wang, and F. Liu, "Data driven chiller sequencing for reducing HVAC electricity consumption in commercial buildings," in *Proc. ACM e-Energy*, 2018, pp. 236–248.

[98] Z. Zheng, Q. Chen, C. Fan, N. Guan, A. Vishwanath, D. Wang, and F. Liu, "An edge based data-driven chiller sequencing framework for HVAC electricity consumption reduction in commercial buildings," *IEEE Trans. Sustain. Comput.*, early access, Jul. 30, 2019, doi: 10.1109/TSUSC.2019.2932045.

[99] B. Yuce and Y. Rezgui, "An ANN-GA semantic rule-based system to reduce the gap between predicted and actual energy consumption in buildings," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 3, pp. 1351–1363, Jul. 2017.

[100] N. K. Thokala, A. Bapna, and M. G. Chandra, "A deployable electrical load forecasting solution for commercial buildings," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2018, pp. 1101–1106.

[101] Y. He, G. J. Mendis, and J. Wei, "Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2505–2516, Sep. 2017.

[102] J. Yan, H. He, X. Zhong, and Y. Tang, "Q-learning-based vulnerability analysis of smart grid against sequential topology attacks," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 200–210, Jan. 2017.

[103] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—A novel pooling deep RNN," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5271–5280, Sep. 2018.

[104] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time EV charging scheduling based on deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Sep. 2019.

[105] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. S. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, Jan. 2018.

[106] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer, and A. Kovacs, "Enhancements of V2X communication in support of cooperative autonomous driving," *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 64–70, Dec. 2015.

[107] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, "Road traffic forecasting: Recent advances and new challenges," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 2, pp. 93–109, Summer 2018.

[108] B. Ghosh, B. Basu, and M. O'Mahony, "Bayesian time-series model for short-term traffic flow forecasting," *J. Transp. Eng.*, vol. 133, no. 3, pp. 180–189, Mar. 2007.

[109] S. Sun and X. Xu, "Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 466–475, Jun. 2011.

[110] Z. Liu, Z. Li, K. Wu, and M. Li, "Urban traffic prediction from mobility data using deep learning," *IEEE Netw.*, vol. 32, no. 4, pp. 40–46, Jul. 2018.

[111] B. P. Gokulan and D. Srinivasan, "Distributed geometric fuzzy multiagent urban traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 714–727, Sep. 2010.

[112] H. Ceylan and M. G. H. Bell, "Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing," *Transp. Res. B, Methodol.*, vol. 38, no. 4, pp. 329–342, May 2004.

[113] F. Cicirelli, A. Guerrieri, G. Spezzano, and A. Vinci, "An edge-based platform for dynamic smart city applications," *Future Gener. Comput. Syst.*, vol. 76, pp. 106–118, Nov. 2017.

[114] X. He, K. Wang, H. Huang, and B. Liu, "QoE-driven big data architecture for smart city," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 88–93, Feb. 2018.

[115] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent edge computing for IoT-based energy management in smart cities," *IEEE Netw.*, vol. 33, no. 2, pp. 111–117, Mar. 2019.

[116] M. Mohammadi and A. Al-Fuqaha, "Enabling cognitive smart cities using big data and machine learning: Approaches and challenges," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 94–101, Feb. 2018.

[117] H. Jiang, C. Cai, X. Ma, Y. Yang, and J. Liu, "Smart home based on WiFi sensing: A survey," *IEEE Access*, vol. 6, pp. 13317–13325, 2018.

[118] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proc. ACM MobiCom*, 2013, pp. 27–38.

[119] W. Kleiminger, S. Santini, and F. Mattern, "Smart heating control with occupancy prediction: How much can one save?" in *Proc. ACM Ubicomp*, 2014, pp. 947–954.

[120] A. Vishwanath, V. Chandan, C. Mendoza, and C. Blake, "A data driven pre-cooling framework for energy cost optimization in commercial buildings," in *Proc. 8th Int. Conf. Future Energy Syst. (e-Energy)*, 2017, pp. 157–167.

[121] F.-J. Ferrández-Pastor, H. Mora, A. Jimeno-Morenilla, and B. Volckaert, "Deployment of IoT edge and fog computing technologies to develop smart building services," *Sustainability*, vol. 10, no. 11, p. 3832, 2018.

[122] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Ann.*, vol. 65, no. 1, pp. 417–420, 2016.

[123] L. Li, K. Ota, and M. Dong, "Deep learning for smart industry: Efficient manufacture inspection system with fog computing," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4665–4673, Oct. 2018.

[124] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional bi-directional LSTM networks," *Sensors*, vol. 17, no. 2, p. 273, 2017.

[125] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018.

[126] F. Wang, X. Fan, F. Wang, and J. Liu, "Backup battery analysis and allocation against power outage for cellular base stations," *IEEE Trans. Mobile Comput.*, vol. 18, no. 3, pp. 520–533, Mar. 2019.

[127] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. IEEE HotWeb*, Nov. 2015, pp. 73–78.

[128] W. Shuai, P. Maille, and A. Pelov, "Charging electric vehicles in the smart city: A survey of economy-driven approaches," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2089–2106, Aug. 2016.

[129] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *J. Manuf. Syst.*, vol. 48, pp. 144–156, Jul. 2018.

[130] F. Wang, F. Wang, X. Fan, and J. Liu, "BatAlloc: Effective battery allocation against power outage for cellular base stations," in *Proc. ACM e-Energy*, 2017, pp. 234–241.

[131] G. Kamath, P. Agnihotri, M. Valero, K. Sarker, and W.-Z. Song, "Pushing analytics to the edge," in *Proc. IEEE GLOBECOM*, Dec. 2016, pp. 1–6.

[132] J. Kone ný, H. Brendan McMahan, F. X. Yu, P. Richtárik, A. Theertha Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*. [Online]. Available: http://arxiv.org/abs/1610.05492

[133] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," 2017, *arXiv:1712.01887*. [Online]. Available: http://arxiv.org/abs/1712.01887

[134] W. Yang Bryan Lim, N. Cong Luong, D. Thai Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," 2019, *arXiv:1909.11875*. [Online]. Available: http://arxiv.org/abs/1909.11875

[135] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," in *Proc. ACM SIGARCH*, 2017, vol. 45, no. 1, pp. 615–629.

[136] Y. Huang, F. Wang, F. Wang, and J. Liu, "DeePar: A hybrid device-edge-cloud execution framework for mobile deep learning applications," in *Proc. IEEE INFOCOM WKSHPS*, Apr. 2019, pp. 892–897.

[137] S. Teerapittayanon, B. McDanel, and H. T. Kung, "BranchyNet: Fast inference via early exiting from deep neural networks," in *Proc. IEEE ICPR*, Dec. 2016, pp. 2464–2469.

[138] L. Du, Y. Du, Y. Li, J. Su, Y.-C. Kuan, C.-C. Liu, and M.-C.-F. Chang, "A reconfigurable streaming deep convolutional neural network accelerator for Internet of Things," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 1, pp. 198–208, Jan. 2018.

[139] S. Jiang, D. He, C. Yang, C. Xu, G. Luo, Y. Chen, Y. Liu, and J. Jiang, "Accelerating mobile applications at the network edge with software-programmable FPGAs," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 55–62.

[140] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems," 2015, *arXiv:1512.01274*. [Online]. Available: http://arxiv.org/abs/1512.01274

FANGXIN WANG (Student Member, IEEE) received the B.S. degree from the Department of Computer Science of Technology, Beijing University of Post and Telecommunication, Beijing, China, in 2013, and the M.S. degree from the Department of Computer Science and Technology, Beijing, in 2016. He is currently pursuing the Ph.D. degree with the School of Computing Science, Simon Fraser University, Burnaby, Canada. His research interests include the Internet of Things, wireless networking, multimedia systems, big data analysis, and machine learning.

MIAO ZHANG received the B.S. degree from the Department of Computer Science of Technology, Sichuan University, Chengdu, China, in 2015, and the M.S. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2018. She is currently pursuing the Ph.D. degree with the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada. Her research interests include cloud computing and multimedia systems.

XIANGXIANG WANG received the B.S. degree from the Department of Computer Science and Technology, East China University of Science and Technology, Shanghai, China, in 2014, and the M.S. degree from the Department of Computer Application, Tsinghua University, Beijing, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada. His research interests include network quality of service, wireless networking, and machine learning.

XIAOQIANG MA received the B.E. degree from the Huazhong University of Science and Technology, China, in 2010, and the M.Sc. and Ph.D. degrees from Simon Fraser University, Canada, in 2012 and 2015, respectively. His research interests include wireless networks, social networks, and cloud computing.

JIANGCHUAN LIU (Fellow, IEEE) received the B.Eng. *(cum laude)* degree in computer science from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree in computer science from The Hong Kong University of Science and Technology, in 2003.

He is currently a Full Professor (with University Professorship) with the School of Computing Science, Simon Fraser University, BC, Canada. He is a Fellow of the Canadian Academy of Engineering and the NSERC E.W.R. Steacie Memorial Fellow. He is a Steering Committee Member of IEEE Transactions on Mobile Computing. He was a co-recipient of the Test of Time Paper Award of the IEEE INFOCOM, in 2015, the ACM TOMCCAP Nicolas D. Georganas Best Paper Award, in 2013, and the ACM Multimedia Best Paper Award, in 2012. He is an Associate Editor of the IEEE/ACM Transactions on Networking, the IEEE Transactions on Big Data, and the IEEE Transactions on Multimedia.