CCF Transactions on Networking
https://doi.org/10.1007/s42045-018-0003-0

REGULAR PAPER

# Exploiting the edge power: an edge deep learning framework

Xiaoyi Fan[1] · Yutao Huang[2] · Xiaoqiang Ma[3] · Jiangchuan Liu[2] · Victor C. M. Leung[1]

Received: 25 July 2018 / Accepted: 22 October 2018

**Abstract**
Crowdsourced big data from Internet users have long been of interest to modern machine learning technologies, and recent advances in *deep learning* have shown great potentials in exploring the hidden information therein. Deep learning relies on strong computation power to process the massive amount of data, which is typical offered by modern data centers, so for data storage. A cloud built on top of the data center, which seamlessly integrates storage and computation, seems to be an ideal platform for learning. It however faces significant challenges from data collection and service distribution over the network, given the end users are globally and remotely distributed. In this article, we present *edge learning* for networked intelligent applications, which complements the cloud-centric design to effectively reduce network traffic and inference latency. We discuss the key design issues of edge learning, including strategies to push data pre-processing and preliminary learning to the network edge, as well as to confine computation to local regions with high accuracy. A prototype demonstrates its feasibility with off-the-shelf hardware and confirms its superiority with realworld experiments.

**Keywords** Edge computing · Deep learning

## 1 Introduction

With the advances in personal computing devices and the deep penetration of mobile broadband networks, the world has been well connected. Globally distributed end users and terminal devices are now actively engaged in the Internet ecosystem for data contribution, rather than consuming data only. The data generated and contributed by them, ranging from text, voice, to picture and video, are enormous and contain invaluable infor-mation worth discovering. Recent advances in *deep neural networking* or *deep learning* have shown great potentials in exploring the hidden information therein, e.g., for speech recognition, video classification, and online shopping recommendation, to name but a few Schmidhuber (2015).

Deep learning relies on strong computation power to process the massive amount of data, which is typical offered by machine clusters, or more general, modern data centers. A data center also serves as the *rendezvous* of the data worldwide. A cloud, built on top of data centers, further offers virtualized and auto-scaled compu-tation and storage resources, providing a seemingly ideal platform for learning. For instance, the Deep Learning AMIs (Amazon Machine Images) in the AWS cloud provide machine learning practitioners and researchers with the infrastructure and tools to accelerate learning at any scale. Amazon EC2 instances can be launched with pre-installed popular deep learning frameworks, e.g., TensorFlow and Caffe, to train sophisticated AI models.

While the cloud-centric learning works well for the data that are available in datacenters, gathering the data to a data center however involves transmission over the Internet.

✉ Jiangchuan Liu
jcliu@sfu.ca

Xiaoyi Fan
xiaoyif@ece.ubc.ca

Yutao Huang
yutaoh@sfu.ca

Xiaoqiang Ma
mxqhust@gmail.com

Victor C. M. Leung
vleung@ece.ubc.ca

[1] Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada

[2] School of Computing Science, Simon Fraser University, Burnaby, Canada

[3] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China

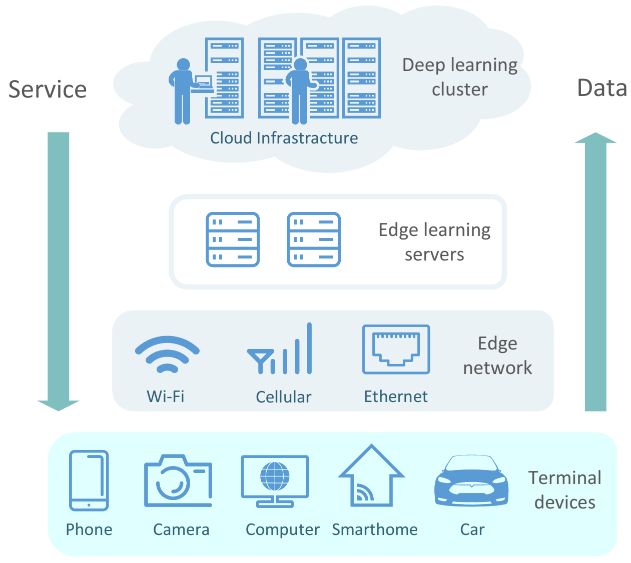


**Fig. 1** The overview of edge computing

The transmission from data sources worldwide unavoidably incurs high traffic and, more importantly, latency that challenges such realtime learning applications as face recognition and human tracking in camera networks. For these applications, their raw data can be huge, which takes relatively long time for both data transfer and model training; their ultra-low inference latency requirement, say less than 10 ms, can be hardly met by the cloud-centric paradigm, since it can be quite difficult for today's Internet to sustain a latency under 100 ms, even with ultra high-speed link and distributed data centers. The highly diversified data sources and distribution channels (e.g., WiFi and 4G/5G) further complicate the design, deployment, and management among the data sources, the learning engine, and the end users.

The concept of *edge computing* has been recently advocated as a complement to cloud computing. It pushes applications, data, and computing content away from the centralized data centers. Extending services to the network edge, a substantial amount of storage, communication, control, configuration, measurement, and management can be placed close to end users, as shown in Fig. 1. As such, it can significantly accelerate the training process by reducing the traffic transferred to the cloud and the inference latency for a broad spectrum of deep learning applications.

In this article, we present *edge learning* for networked intelligent applications, which complements the cloud-centric design to effectively reduce the network traffic and inference latency for deep learning. We present details on how to push data pre-processing and feature extracting to the network edge. We have implemented a practical edge learning prototype and showcase with an image analytic

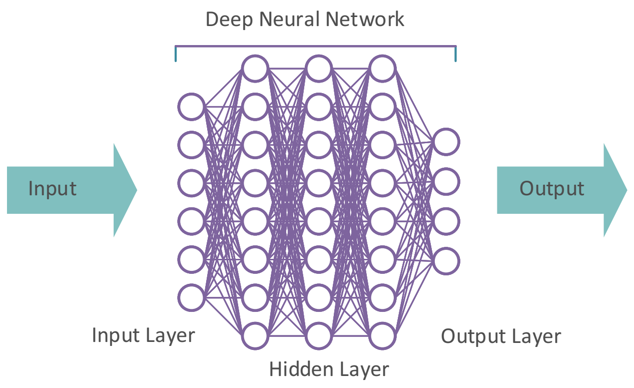


**Fig. 2** Deep neural networks

application that recognizes human face with the MS-Celeb-1M dataset.[1] Our experiments show that, with comparable learning accuracy as the cloud-centric design, edge learning reduces the inference latency and network traffic volume by 69% and 80%, respectively.

## 2 Deep learning over the network

### 2.1 Deep learning primer

Neural network approaches, e.g., feed-forward neural networks and recurrent neural networks, have attracted significant efforts from the research community in recent decades, Fig. 2 shows feed-forward neural networks Huang et al. (2004), a new emerging generation of machine learning technique, which can be divided into the three main layers: input layer, hidden layer, and output layer. Perceptrons are arranged in layers, with the first layer taking in inputs and the last layer producing outputs. The hidden layers have no connection with the external world, and each perceptron in one layer is connected to every perceptron on the next layer, so that the information is fed forward from one layer to the next. Recent years, multi-layer neural network Schmidhuber (2015), namely deep neural networks or deep learning, has gained huge success in efficiently solving a broad spectrum of problems, such as image recognition, computer vision, natural language processing, and speech recognition. Deep learning also has been widely used in the industry, e.g., Google, Microsoft and Nvidia. In particular, Google has adopted deep learning for the speech and image

[1] MS-Celeb-1M is a large scale real world face image dataset to the public, encouraging researchers to develop the best face recognition techniques to recognize one million human face entities identified from Freebase. http://www.msceleb.org/.

recognition capabilities of Google Translate and Google Photos, respectively.

Two recent trends have sparked widespread use of deep learning: the powerful and efficient parallel computing provided by GPU computing, and the avail-ability of massive amounts of training data. With thou-sands of computational cores, GPUs have become the processor of choice for processing big data for data scientists. For instance, Google's TensorFlow and DeepMind have exploited GPUs to parallelize deep network training, which enables to train the deep neural networks using far larger training sets.

## 2.2 Deep learning on cloud

Cloud-based deep learning is a promising practical solution to leveraging dynamic streaming data for online training. For example, Amazon EC2 with Elastic GPUs[2] allows users to run deep learning applications in the cloud. The state-of-the-art cloud-based deep learning system, e.g., face++ for face detection,[3] forwards massive geologically distributed data to data centers, and then delivers the results to users to enable good user experience. As mentioned, the deep learning service relies on the availability of massive amounts of training data. The advances in cloud computing offer the preferred choice in this era of big data computation at terabyte and even petabyte-scales, which inevitably requires not only GPU computing resources but also high network bandwidth.

Putting all the computing tasks on the cloud, especially deep learning applications, has been proven to be an viable way for data processing in certain scenarios, since the computation power on the cloud outclasses the capability of end user devices. However, compared to the fast developing data processing speed, the bandwidth of the network has come to a standstill. With the growing volume of data generated by end users, the speed of data transportation is becoming the bottleneck for the cloud-based deep learning paradigm.

We take an image analytic application as an example, which can real-time search a specific person in the city with the widespread of mobile phones and network cameras. The video analytic application not only requires huge GPU resources for computation, but also tremendous network resources for video transmission. Yet, the cloud-based deep learning paradigm is difficult to leverage the wide area camera data, since the videos from the camera are extremely time-consuming to be uploaded to the cloud. Even if the data are uploaded to the cloud, deep learning takes a long time to process such a huge quantity of data, which is not tolerable for searching a person. Thus, the state-of-the-art cloud computing is no longer suitable for video analytics with the large data transmission latency and volume.

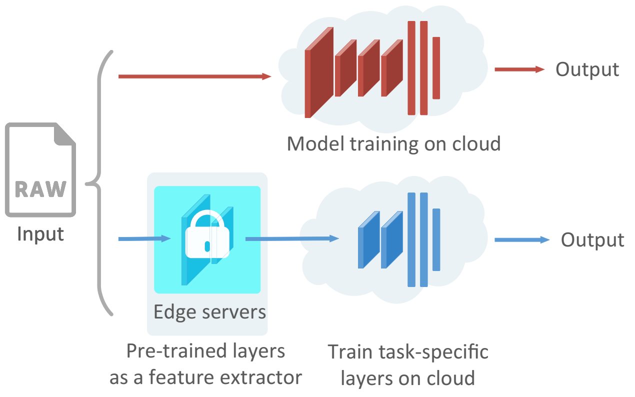


**Fig. 3** From cloud learning to edge learning

## 2.3 From cloud to edge learning

Network as a bridge between diverse end-user devices and cloud computing platforms certainly open new opportunities. *Edge networks*, which are directly connected to end-user devices, play a critical role here in aggregating such distributed data and forwarding them to data centers. Edge computing refers to the enabling technologies that allow computation to be performed at the edge networks, which aims to reduce latency by bringing the computation and storage resources from cloud infras-tructures to the proximity of data sources, e.g., edge computation offloading Tan et al. (2017) Li et al. (2017) Chen et al. (2016) Habak et al. (2015), edge caching Drolia et al. (2017) Huang et al. (2017), and edge resource allocation Wang et al. (2017). The edge computing devices can be any computing or networking resource residing between data sources and cloud-based data centers, e.g., a 5G cellular tower between smartphones and the cloud infrastructures.

Deep learning has achieved great success in numerous applications. Yet, the cloud-base services are still far away from satisfaction, which also has attracted significant efforts from the network prespective. DeepCham Li et al. (2016) is proposed as an adaptive mobile object recognition framework, which introduces an edge master server to coordinate with participating mobile users and collabo-ratively train a domain-aware adaptation model. A distributed deep neural network (DDNN) Teerapittayanon et al. (2017) is proposed to partition neural networks across mobile devices, edges, and cloud, so as to reduce the latency.

In this article, we propose *edge learning* as shown in Fig. 3, which concerns not only data transmission, but rather the life cycle of deep learning, so as to provide ultra-low latency application services. Edge learning introduces

[2] https://aws.amazon.com/ec2/elastic-gpus/.

[3] https://www.faceplusplus.com/.

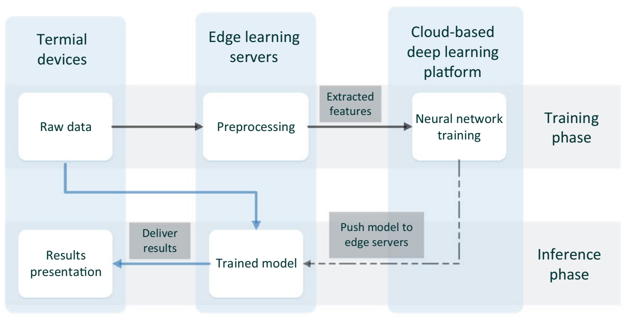


**Fig. 4** Edge learning framework

computation offloading on edge servers to minimize the inference latency, as well as the traffic volume. Instead of training the whole deep learning model in the cloud infrastructures, part of the model training is pushed to the edge servers; the extracted features are delivered to cloud server for further model training, and the results are returned to end users after successfully processing.

With the edge learning paradigm, the request of searching a person can be generated from the cloud and pushed to tasks in a target area. Each edge servers can perform the request and report the extracted features from videos, so as to alleviate bandwidth pressure and minimize the inference latency. Also, the cloud platform can determine whether to query the raw videos based on extracted features.

## 3 The edge learning framework design

In this section, we propose *edge learning*, a real-world case to discuss the challenges and solutions in designing and implementing an edge computing system for deep learning applications, which is a complementary concept to existing cloud infrastructure.

### 3.1 Architecture and design

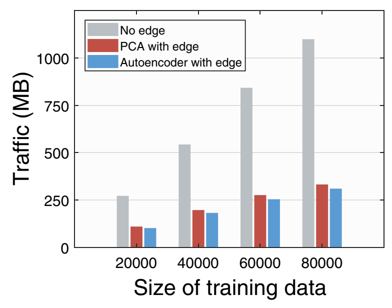
Figure 4 provides a high level description of the edge learning workflow, which consists of three major framework: end-user devices, edge learning servers, and cloud-based deep learning platform. In the edge learning framework, terminal devices, e.g., mobile phones, cameras, and IoT devices, generate a large volume data, such as images streaming from the smart phone camera, and then upload them to edge servers. The workflow of our edge learning framework consists two phases, including training phase and inference phase.

In the training phase of our framework, a preprocessing module is deployed on edge servers. The basic idea is feature extractors are pushed from cloud towards the edge servers to reduce the traffic volume. The edge learning servers gather the massive raw data from end users and perform pre-processing with an autoencoder Bengio (2009) or a principal component analysis (PCA) Wold et al. (1987) algorithm, so as to filter out the noises and extract key features. We take the autoencoder as an example to illustrate the effectiveness of our edge learning framework. The autoencoder is a data compression algorithm, which can abstract visual features such as edges and curves from raw images. The weights of the autoencoder on pre-processing module have been pre-trained based on a large dataset. Unless the applications on cloud infrastructures have a very unique problem space and dataset, the neural networks to be executed on the cloud will only need to detect the similar visual features and modify the input layer's dimension according to the autoencoder's configuration without changing other structure. Thus, we can use the pre-trained auto-encoder on edge servers to preprocess them and upload the extracted features to the cloud-based deep learning platform. The network bandwidth demands from end users to cloud are significantly reduced, which alleviate the pressure on the network, compared to state-of-the-art cloud computing architecture. As Fig. 4 illustrates, the cloud-based deep learning platform, equipped with powerful and scalable GPU resources, train the deep network based on the extracted features from edge servers.

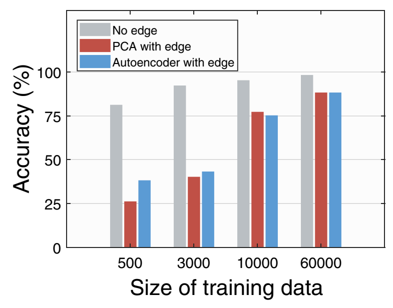
In the inference phase of deep learning applications, edge learning framework pushes the trained model on edge servers to provide real-time services, which can significantly reduce the inference latency. The learning-based applications that have an ever stricter latency requirement, e.g., automated driving cars and virtual reality, demand lightning-fast deep learning inference within tens of milliseconds. Edge learning can bringing inference acceleration with a ultra-low latency, which offloads inference computation from the cloud platform to edge servers.

### 3.2 Implementation on our testbed

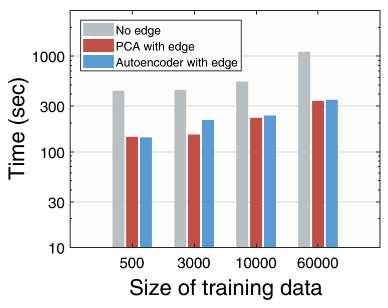
We have implemented edge learning design in a real-world testbed. Our testbed consists of terminal devices, one edge servers, and the cloud-based deep learning platform. In particular, the terminal devices include 4 Google Nexus 9 Android Tablets. The edge server works on 2 Dell servers (OPTIPLEX 7010), each equipped with an Intel Core i7-3770 3.4 GHz quad core CPU, 16 GB 1333 MHz DDR3 RAM, and an NVIDIA GeForce GTX 960 GPU. The cloud-based deep learning platform consists of one high performance customized PC, which is equipped with



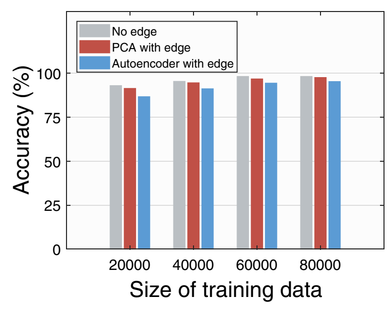
**(a)** Traffic



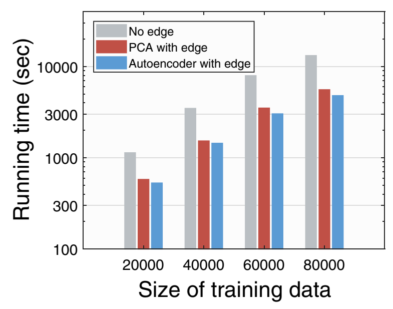
**(b)** Training Accuracy (15 Epochs)



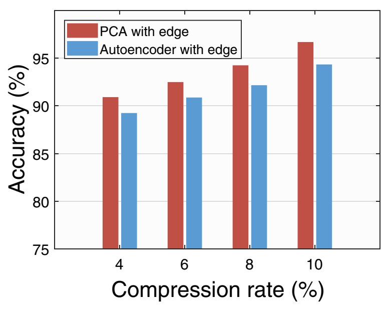
**(c)** Training Time (15 Epochs)



**(d)** Training Accuracy (200 Epochs)



**(e)** Training Time (200 Epochs)



**(f)** Accuracy with different compression rate

**Fig. 5** Performance evaluation. **a** Traffic. **b** Training Accuracy (15 Epochs). **c** Training time (15 Epochs). **d** Training Accuracy (200 Epochs). **e** Training Time (200 Epochs). **f** Accuracy with different compression rate

an Intel Core i7-6850K 3.60 GHz 12-core CPU and Dual Nvidia GTX 1080 Ti GPUs.

In the training phase, we emulate the crowdsourced mechanism. The end devices upload the images from MS-Celeb-1M dataset[4] with the ground truth of the celebrities label in a sequence to the edge server. The edge server performs the autoencoder and the principle component analysis (PCA) with the received images, and then uploads the extracted features to the cloud-based deep learning platform. The cloud-based deep learning platform trains the deep learning model based on extracted features, which is implemented in Keras[5] with Tensorflow[6] backend.

In the inference phase, our edge learning app running on smart phones can timely upload the images captured by the device's camera to edge servers, and identify the celebrities based on the pre-trained deep learning architecture, so as to provide realtime services to users.

[4] http://www.msceleb.org/.

[5] https://keras.io/.

[6] https://www.tensorflow.org/.

## 4 Evaluation

In this section, we now evaluate the effectiveness of the proposed edge learning system. We evaluate the system with two different preprocessing schemas on edge servers, e.g., PCA and autoencoder, to compare with the basic Keras system, in terms of the training time, learning accuracy, and traffic size from edge to cloud. We use a public MS-Celeb-1M dataset, which is a large training dataset which covers the top 100K celebrities. In our experiment, we choose 1,000 celebrities with 100,000 images and 1,089 MB volume, where 80% of the data are used as a training set and the remaining 20% are used as a test set. We run a representative application to evaluate the performance of these systems, which recognizes human face images with the existing information.

We first investigate the generated traffic uploaded to the deep learning cluster. From Fig. 5a, we can figure out that performing PCA on the edge can reduce 60%-70% traffic transmitted to the cluster, and autoencoder can achieve a 63%-72% reduction on the traffic size. The benefits come from that PCA and autoencoder compress the raw image data and reduce the dimensionality of the input data to neural networks.

Then we set the training epoch number as 15 and explore the running time and the accuracy for training a CNN model on the deep learning cluster. As Fig. 5b shows, though the accuracy is relatively low when the training data scale is small, the accuracy grows fast as the training data scale increases. When the data set scale comes to 60,000, our edge learning framework has 10% less accuracy than direct learning. Fig. 5c shows that our edge learning framework significantly reduces the running time by 65%, as compared to the state-of-the-art cloud-based deep learning system without edge servers. The benefits stem from preprocessing on input data with the compressing functionality of PCA and autoencoder.

Then we increase the training epoch number to 200. Fig. 5d shows that edge learning framework with the PCA can achieve similar accuracy with the state-of-the-art cloud-based deep learning system, while autoencoder gets slightly lower accuracy. We can see that with more data for the model training, our edge learning framework can achieve a higher accuracy. When the number of training epoch number increases, our edge learning framework can achieve a similar accuracy with the state-of-the-art cloud-based deep learning system. On the other hand, our approach achieves a huge reduction in training time, as shown in Fig. 5e, although the overall running time still raises due to the increasing epoch number. Fig. 5f illustrates that for both PCA and autoencoder, the accuracy will increase when the dimension increases. Since the compression rate affects the accuracy, this result indicates that there should be a trade-off between the accuracy and the compression rate.

## 5 Future work

Based on the proposed edge learning framework, the following future work can be further pursued.

### 5.1 Learning techniques on edge servers

We will implement our edge learning framework for a large scale distributed camera network, which consists of hundreds of smart cameras as end devices. The challenge is that the large-scale raw data produced by the camera network will require sophisticated compressing and learning techniques on the edge to provide cost-effective and low-latency data acquisition. Also, the data collected by the cameras in close proximity will have spatial and temporal relations, which are to be exploited on the edge servers to realize regional learning, so as to improve the performance of our edge learning framework.

### 5.2 Enabling customer-provided resources for edge learning

Driven by the strong demands, industrial pioneers have offered commercial cloud platforms for fast deployment of deep learning environments, which are known to be powerful and effective. Yet the cloud customers are normally pure consumers of cloud resources, which means that their local resources, though abundant, have been largely ignored. We can seamlessly integrate customers' local resources as spot nodes into edge networks, enabling them to sell, buy, and utilize these resources. We can also investigate the potentials and challenges toward enabling customer-provided resources for cloud-based deep learning.

### 5.3 Privacy and data collection

Users would be concerned about the privacy risk of sharing their personal mobile data with a service server. Thus, a low percentage of users will opt out of sharing their personal data, unless trustworthy privacy mechanisms are applied. An alternative research direction is deploying the trained model on the edge nodes to serve users. Meanwhile, anonymized data collection is adopted and the system can encourage the sharing of mobile data in return for rewarding points.

## 6 Conclusion

In this article, we presented a retrospective view of past and present edge computing, followed by the very recent advances in crowdsourcing and deep learning. We presented the design principles of edge computing with deep learning. The unique challenges therein were discussed, particularly in pushing the tasks to edge servers and designing a framework for deep learning. In the case study, we developed an edge learning system and illustrate its advantages over state-of-the-art cloud-based counterpart.

## Compliance with ethical standards

**Conflict of Interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

## References

Bengio, Y., et al.: Learning deep architectures for AI. Found. Trends® Mach. Learn. **2**(1), 1–127 (2009)

Chen, X., Jiao, L., Li, W., Fu, X.: Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Trans. Netw. **5**, 2795–2808 (2016)

Drolia, U., Guo, K., Tan, J., Gandhi, R., Narasimhan, P.: Cachier: Edge-caching for recognition applications. In: IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 276–286 (2017)

Habak, K., Ammar, M., Harras, K.A., Zegura, E.: Femto clouds: Leveraging mobile devices to provide cloud service at the edge. In: IEEE International Conference on Cloud Computing (CLOUD), pp. 9–16 (2015)

Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE International Joint Conference on Neural Networks (IJCNN), vol. 2, pp. 985–990 (2004)

Huang, Y., Song, X., Ye, F., Yang, Y., Li, X.: Fair Caching Algorithms for Peer Data Sharing in Pervasive Edge Computing Environments. In: IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 605–614 (2017)

Li, D., Salonidis, T., Desai, N.V., Chuah, M.C.: Deepcham: Collaborative edge-mediated adaptive deep learning for mobile object recognition. In: IEEE/ACM Symposium on Edge Computing (SEC), pp. 64–76 (2016)

Li, T., Magurawalage, C.S., Wang, K., Xu, K., Yang, K., Wang, H.: On efficient offloading control in cloud radio access network with mobile edge computing. In: IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 2258–2263 (2017)

Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)

Tan, H., Han, Z., Li, X.Y., Lau, F.C.: Online job dispatching and scheduling in edge-clouds. In: IEEE International Conference on Computer Communications (INFOCOM), pp. 1–9 (2017)

Teerapittayanon, S., McDanel, B., Kung, H.T.: Distributed deep neural networks over the cloud, the edge and end devices. In: IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 328–339 (2017)

Wang, L., Jiao, L., Li, J., Muhlhauser, M.: Online resource allocation for arbitrary user mobility in distributed edge clouds. In: IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 1281–1290 (2017)

Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemometr. Intell. Lab. Syst. **2**(1–3), 37–52 (1987)

**Xiaoyi Fan** received the B.E. degree from Beijing University of Posts and Telecommunications in 2013, the M.Sc. degree from Simon Fraser University in 2015, and the Ph.D. degree from Simon Fraser University in 2018. He is now a postdoctoral research fellow in the Department of Electrical and Computer Engineering at The University of British Columbia, Canada. His areas of interest are Internet-of-Things, big data and deep learning.

**Yutao Huang** received the B.Eng degree from Shanghai Jiaotong University, China, in 2016. He is currently pursuing the M.Sc. degree with the School of Computing Science, Simon Fraser University, British Columbia, Canada. His areas of interest are machine learning, edge computing, and cloud computing.

**Xiaoqiang Ma** received the B.Eng degree from Huazhong University of Science and Technology, China, in 2010. He received the M.Sc. and Ph.D. degrees from Simon Fraser University, Canada, in 2012 and 2015, respectively. His areas of interest are wireless networks, social networks, and cloud computing.

**Jiangchuan Liu** received the B.Eng. degree (Cum Laude) from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree from The Hong Kong University of Science and Technology in 2003, both in Computer Science. He is currently a Full Professor (with University Professorship) in the School of Computing Science at Simon Fraser University, British Columbia, Canada. He is an IEEE Fellow and an NSERC E.W.R. Steacie Memorial Fellow. He is a Steering Committee Member of IEEE Transactions on Mobile Computing, and Associate Editor of IEEE/ACM Transactions on Networking, IEEE Transactions on Big Data, and IEEE Transactions on Multimedia. He is a co-recipient of the Test of Time Paper Award of IEEE INFOCOM (2015), ACM TOMCCAP Nicolas D. Georganas Best Paper Award (2013), and ACM Multimedia Best Paper Award (2012).

**Victor C. M. Leung** received the B.A.Sc. (Hons.) degree in electrical engineering from the University of British Columbia (UBC) in 1977, and was awarded the APEBC Gold Medal as the head of the graduating class in the Faculty of Applied Science. He attended graduate school at UBC on a Canadian Natural Sciences and Engineering Research Council Postgraduate Scholarship and received the Ph.D. degree in electrical engineering in 1982. From 1981 to 1987, Dr. Leung was a Senior Member of Technical Staff and satellite system specialist at MPR Teltech Ltd., Canada. In 1988, he was a Lecturer in the Department of Electronics at the Chinese University of Hong Kong. He returned to UBC as a faculty member in 1989, and currently holds the positions of Professor and TELUS Mobility Research Chair in Advanced Telecommunications Engineering in the Department of Electrical and Computer Engineering. Dr. Leung has co-authored more than 1200 journal articles, conference papers, and book chapters, and co-edited 14 book titles. Several of his papers had been selected for best paper awards. His research interests are in the broad areas of wireless networks and mobile systems. Dr. Leung is a registered Professional Engineer in the

Province of British Columbia, Canada. He is a Fellow of the Royal Society of Canada, the Engineering Institute of Canada, and the Canadian Academy of Engineering. He was a Distinguished Lecturer of the IEEE Communications Society. He is serving on the editorial boards of the IEEE Transactions on Green Communications and Networking, IEEE Transactions on Cloud Computing, IEEE Access, IEEE Network, Computer Communications, and several other journals, and has previously served on the editorial boards of the IEEE Journal on Selected Areas in Communications – Wireless Communications Series and Series on Green Communications and Networking, IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, IEEE Transactions on Computers, IEEE Wireless Communications Letters, and Journal of Communications and Networks. He has guest-edited many journal special issues, and provided leadership to the organizing committees and technical program committees of numerous conferences and workshops. He received the IEEE Vancouver Section Centennial Award, the 2011 UBC Killam Research Prize, the 2017 Canadian Award for Telecommunications Research, the 2018 IEEE TGCC Distinguished Technical Achievement Recognition Award, and the 2018 ACM MSWiM Reginald Fessenden Award. He co-authored papers that won the 2017 IEEE ComSoc Fred W. Ellersick Prize, the 2017 IEEE Systems Journal Best Paper Award, and the 2018 IEEE CSIM Best Journal Paper Award.