BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 04/06 pp222–233
Volume 1, Number 3, September 2018
DOI: 10.26599/BDMA.2018.9020020

QoE-Driven Big Data Management in Pervasive Edge Computing Environment

Qianyu Meng, Kun Wang*, Xiaoming He, and Minyi Guo

Abstract: In the age of big data, services in the pervasive edge environment are expected to offer end-users better Quality-of-Experience (QoE) than that in a normal edge environment. However, the combined impact of the storage, delivery, and sensors used in various types of edge devices in this environment is producing volumes of high-dimensional big data that are increasingly pervasive and redundant. Therefore, enhancing the QoE has become a major challenge in high-dimensional big data in the pervasive edge computing environment. In this paper, to achieve high QoE, we propose a QoE model for evaluating the qualities of services in the pervasive edge computing environment. The QoE is related to the accuracy of high-dimensional big data and the transmission rate of this accurate data. To realize high accuracy of high-dimensional big data and the transmission of accurate data through out the pervasive edge computing environment, in this study we focused on the following two aspects. First, we formulate the issue as a high-dimensional big data management problem and test different transmission rates to acquire the best QoE. Then, with respect to accuracy, we propose a Tensor-Fast Convolutional Neural Network (TF-CNN) algorithm based on deep learning, which is suitable for high-dimensional big data analysis in the pervasive edge computing environment. Our simulation results reveal that our proposed algorithm can achieve high QoE performance.

Key words: Quality-of-Experience (QoE); high-dimensional big data management; deep learning; pervasive edge computing

1 Introduction

Various kinds of edge devices, including mobile phones, iPads, laptops, connected vehicles, smart cameras, and a range of Internet-of-Things (IoT) devices[1, 2] have been deployed in the pervasive edge computing environment, which refers to the nearest edge sources of content and data that offer smart services[3, 4]. These edge devices possess communication, sensing, computing, and storage capacities. As a result, they produce pervasive and ever-increasing volumes of big data regarding physical phenomena in the pervasive edge computing environment, which results in the massive scope of big data evolving from the gigabyte to the exabyte. The produced data is also referred to high-dimensional big data[5, 6]. When extracted from these data in the pervasive edge computing environment, the enormous amount of accurate data can improve the Quality-of-Experience (QoE)[7] provided by big data services, since end-user social groups expect high accuracy and data transmission rates than are available in the normal edge environment.

The QoE concept is a well-known measurement

• Qianyu Meng and Xiaoming He are with the Jiangsu Engineering Research Center of Communication and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China. E-mail: isqy.meng@gmail.com; isxmhe@gmail.com.
• Kun Wang is with the Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing University of Posts and Telecommunications, Nanjing 210003, and the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: kwang@njupt.edu.cn.
• Minyi Guo is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: guomy@cs.sjtu.edu.cn.
∗ To whom correspondence should be addressed.
Manuscript received: 2018-02-01; accepted: 2018-02-14

mechanism for determining the overall perception of the quality-of-service (QoS)[8,9], i.e., the evaluation of QoS as experienced by end-users. Therefore, both academic and industry researchers have shifted their attention from QoS parameters like jitter, throughput, packet loss, and delay to the concept of QoE. The International Telecommunication Union has defined the QoE concept as *the entire thing of availability of services subjectively perceived by end-users*. The definition of QoE by the European Qualinet is the degree of satisfaction or annoyance of the end-users of services because the utility and/or the expectations regarding services are based on end-user attitudes and current situations[7]. In summary, the common understanding of QoE is as follows: QoE is a new measurement for edge computing services which is based on vital parameters.

Recently, many QoS-based methods have been developed to optimize the efficiency and performance of the whole environment, as proposed in Refs. [8, 9]. Even though the parameters of QoS offer good objective measurement criteria, they can not directly determine the quality of end-user perceptions. QoE, in contrast, can refer to both the performance and efficiency of services as measured by QoS, as well as the subjective opinions of end-users. Therefore, QoE is more suitable with respect to end-users than is QoS.

To date, many researchers have devoted their efforts to high-dimensional big data management in the pervasive edge environment with respect to QoE. However, it is not a simple matter to quickly train the accuracy of high-dimensional big data and establish an effective transmission rate of accurate data for big data services with respect to QoE in the pervasive edge computing environment, which may contain some constraints, e.g., volume, variety, dimension, bandwidth, etc. Compared with traditional methods, machine learning[10,11] techniques have some unique advantages in the extraction of big data and many studies have applied deep learning[12,13] techniques in the pervasive edge computing environment. A typical example of a straightforward solution to achieving state-of-the-art accuracy in high-dimensional big data analysis is the use of a Convolutional Neural Network (CNN) technique[14,15], such as image/video processing, speech recognition, or natural language processing. Another option is to use Deep CNN (DCNN)[16] to perform high-dimensional big data analysis, which yields higher accuracy than CNN. DCNN with a tensor (TCNN)[17] is also used to obtain satisfactory accuracy in the analysis of high-dimensional big data.

To improve training speed, Fast Region-based CNN (FR-CNN) has been proposed for the analysis of high-dimensional big data, although its results are less accurate than those of TCNN[18]. In general, Refs. [14–17] have presented solutions for gradually improving accuracy, and the authors in Ref. [18] were able to increase the training speed using CNN. However, none of these methods can guarantee the accuracy of high-dimensional big data or improved training speed. Our investigations indicate that there is as yet no effective technology for enhancing the data transmission rate and accuracy of big services with respect to QoE. That is, despite the presence of high bandwidths, not all service requirements can be met. Nor can the satisfaction of end-users be guaranteed with respect to their experience.

Motivated by the above facts, in this paper, we focus on the issue of QoE in the pervasive edge computing environment. To achieve effective high-dimensional big data management in this environment, we propose a Tensor-Fast CNN (TF-CNN) algorithm that can guarantee accuracy and increase training speed with high-dimensional data. Then, we address the high-dimensional big data management problem using different accurate data transmission rates to identify which yields the best QoE. Our results indicate that our proposed big data management technique using the TF-CNN algorithm achieves better end-user QoE than existing methods. The major contributions of this paper are as follows:

- In the context of the pervasive edge computing environment, we propose a model to improve the QoE of end-users. Through a comprehensive consideration of the accuracy of high-dimensional big data and corresponding transmission rate, we seek a trade-off between the quality of big data services and the experience of end-users.
- To enhance the QoE in the pervasive edge computing environment, we devise a big data management technique based on the TF-CNN algorithm to solve our proposed QoE-maximization problem. This technique involves a carefully considered trade-off between the accuracy of high-dimensional big data and the training speed.
- We conducted an extensive series of experiments to compare the performance of our method with

those of several existing methods, and the results demonstrate the effectiveness of our proposed method.

The remainder of this paper is organized as follows. In Section 2, we review related works and, in Section 3, we discuss big data services with respect to QoE and formulate the QoE-maximization problem. In Section 4, we propose an algorithm for managing big data. We present our experimental results in Section 5, and draw out conclusions in Section 6.

2 Related Work

2.1 QoE

The common understanding of QoE is that it is a novel measurement technique for use by services and is determined based on the quality of the whole service environments and the experience of end-users. QoE has been applied to a variety of scenarios. For instance, Chen et al.[19] examined the current demands of end-users ranging from transmission technologies to heterogeneous devices and offered a heterogeneous QoE technique that supports a wide variety of multimedia devices critical to video broadcasting in wireless networks. Similarly, Zhao et al.[20] introduced selected issues including QoE modeling of the video transmission point-to-point chain, subjective QoE management and objective QoE monitoring, and the QoE assessment of video transmission in different network features. Kim et al.[21] summarized the latest video transmission technologies with regard to scalable video coding in multiple-input-multiple-output systems with cross-layer designs and proposed unequal error protection solutions with respect to QoE in the delivery of video over massive multiple-input-multiple-output systems with respect to content characteristics. Liang et al.[22] then proposed a novel mechanism for bandwidth provisioning and proactive caching as well as joint adaptive video streaming. This mechanism can enhance caching with respect to QoE in wireless software-defined networks. Lastly, Wang et al.[23] presented a data architecture to enhance personalized QoE in 5G networks and proposed a two-step QoE modeling method that capitalizes on the relationship between end-users and services.

2.2 Big data analysis in pervasive edge computing environment

To the best of our knowledge, only a few studies have investigated the various strategies used in high-dimensional big data analysis. Regarding the intensive computing of massive data by data centers, Ji et al.[24] conducted a wide-ranging study of the MapReduce paradigm based on its low cost, large-scale data parallelism, and ability to analyze fault tolerance in the pervasive edge computing environment. The most popular implementation is the Hadoop framework proposed by Zhao and Methi[25], which allows applications to make large scale clusters and offers transparent reliability as well as data transfer. Shi et al.[26] later determined that the greater is the effect of the CNN algorithm at the edge, the greater accuracy is achieved. Similarly, Zhang et al.[27] considered the accuracy of high-dimensional big data analysis and concluded that most accuracy enhancements are achieved by the use of effective algorithms at the edge alone. However, most studies cannot guarantee the accuracy of big data analysis or increase in training speed. Our proposed algorithm for high-dimensional big data analysis differs from the above methods in that it analyzes high-dimensional big data using a tensor representation model[17] and truncated Singular Value De-composition (SVD)[18] simultaneously to ensure accuracy and increased training speed for high-dimensional big data in the pervasive edge computing environment.

2.3 Management of transmission rate for big data

Transmission rate management methods for big data, such as high bandwidth, have been adopted to address the transmission rate challenges posed by big data. Li and Wang[28] explored a strategy for optimizing bandwidth allocation based on the relationships between satisfaction and data rate, with respect to the delay experienced by the end-user. Similarly, Borujeny et al.[29] studied the effect of pairing on the sum rate and general rate of a multi-way relay channel using a functional decode-forward relaying strategy in which end-users experience asymmetric channel conditions. The authors proposed a graphical model in their pairwise transmission strategy to maximize the data rate. A number of studies[30–33] have proposed methods based on code-shifted differential chaos switch importance that modulate code index to realize a high data transmission rate and high speed transmission scheme that supports data rate faster than 100 Mb/s. In summary, most researchers have focused exclusively on the high bandwidth issue. However, high bandwidth alone cannot meet all service requirements nor enhance the satisfaction of the end-user's experience in the pervasive edge computing environment.

In this paper, we focus on the quality of big data services with respect to QoE in the pervasive edge computing environment. We propose a novel QoE model for big data services and compare the performance of our proposed advanced algorithm in the management of high-dimensional big data with those used in other models.

3 System Model and Problem Formulation

In this section, we propose a QoE model for achieving accuracy in high-dimensional big data as well as an effective and accurate data transmission rate. Then, we formulate the QoE-maximization problem to be investigated.

3.1 System model

Figure 1 shows a schematic high-dimensional big data system in the pervasive edge computing environment. The system components include the data servers, Services Providers (SPs), data analysis units, and end-users. Table 1 lists the important notations we used in this study. The data servers collect the raw data

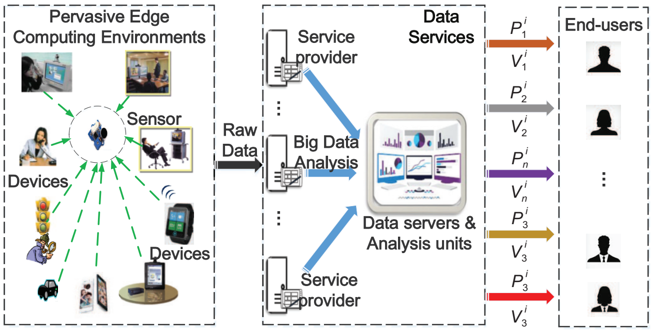

Fig. 1 Big data in the pervasive edge computing environment.

generated by various sensors and edge devices. The data analysis units analyze the data in each time slot i, $i \in \mathbf{N}^+$, and return accurate data. We denote the accuracy of high-dimensional big data by p^i, $i \in \mathbf{N}^+$. End-users can then transmit their requirements and feedback to the SP. As such, the data analysis units continuously adapt their data presentation, taking into account the QoE of the end-users of different services. In this manner, our big data scenario yields accurate data, and transmits this data accurately to end-users to achieve a high QoE. We denote the transmission rate in each time slot i, $i \in \mathbf{N}^+$, of this accurate data by r^i,

Table 1 Important notations.

p^i	Accuracy of high-dimensional big data in each time slot i, $i \in \mathbf{N}^+$
r^i	Transmission rate of accurate data in each time slot i, $r^i \in \mathbf{Q}$, $i \in \mathbf{N}^+$
Q	Set of k division point values of $r^i \in \mathbf{Q}$, $i \in \mathbf{N}^+$
$r^i_{\min}$	Minimum transmission rate in each time slot i, $i \in \mathbf{N}^+$
$r^i_{\max}$	Maximum transmission rate in each time slot i, $i \in \mathbf{N}^+$
ξ	QoE weighting parameter between the accuracy of high-dimensional big data and transmission rate of accurate data
ϕ	QoE-maximization as our formulation
$\sim$	Symbol of equivalence indicating that the value of QoE is equivalent to the value of p^i, $i \in \mathbf{N}^+$
ade(.)	Choice function of r^i referring to the fact that the value of r^i can meet requirements of each end-users, $i \in \mathbf{N}^+$
$U_{\text{TF-CNN}}$	Loss function related to TF-CNN
(x, y)	Tensor object
s_θ	Dimensions of TF-CNN
$\frac{\partial U_{\text{TF-CNN}}(\theta)}{\partial k^{(t)}}$	Partial derivative of TF-CNN concerning θ
α	Learning rate
$k = D, B, \beta, b$	D is the $(M+1)$-order weight tensor, B is the M-order tensor, b is the bias tensor, β is the weight
Q_{FM}	$q_{fm} \times l$ matrix comprising the first l left-singular values of X
$\sum_l$	$l \times l$ diagonal matrix including the top l singular values of X
Q_{IM}	$q_{im} \times l$ matrix comprising the first l right singular matrix of X
z_j	Element of the tensor X
$w^{(t)}_{i1\cdots imf1\cdots fm}$	Weight difference between the unit $f1 \cdots fm$ of layer t and the unit $i1 \cdots im$ of layer $t+1$
$k^{(t)}_{f1\cdots fm}$	Weight of kernel L
$\mathrm{up}(\varepsilon^{(t+1)}_{i1\cdots im})$	Upsampling operation that uses factor sc^i to tile the input element of each dimension
Z	Unsorted sum tree
∇	$\frac{\partial U_{\text{TF-CNN}}(\theta,x,y)}{\partial k^{(t)}}$ $(k = D, B, \beta, b)$

$i \in \mathbf{N}^{+}$.

3.2 QoE model in the pervasive edge computing environment

3.2.1 Measurement of QoE

To improve the quality of the end-user experience in the pervasive edge computing environment, we measure QoE as the quality perceived based on the accuracy of high-dimensional big data, which refers to the value of the data and transmission rate of accurate data generated by the data analysis units. Two methods are commonly used to assess QoE. The first is based on the QoE value, which is related to the accuracy of the high-dimensional big data. The lower the accuracy of the high-dimensional big data, the poorer is the experience of the end-users. The second method is based on the transmission rate of accurate data particularly for different applications whereby, the faster the transmission rate, the better the experience of the end-users. The QoE in the pervasive edge computing environment is jointly influenced by these two factors. In the next section, we describe our assessment methods in detail.

3.2.2 QoE model

We consider the accuracy of high-dimensional big data as one QoE factor in the pervasive edge computing environment. When the accuracy of big data is not high, end-users will demand that it be improved. In the pervasive edge computing environment, the transmission rate r^i, $i \in \mathbf{N}^{+}$, of accurate data in each time slot i for big data services can also be considered as a QoE term. Then, we can define the transmission rate function r^i as $\mathrm{ade}(r^i)$, $i \in \mathbf{N}^{+}$, and thereby obtain the QoE in the pervasive edge computing environment:

$$\mathrm{QoE}(p^i, r^i) = a \cdot e^{d \cdot p^i} + \xi \mathrm{ade}(r^i) \tag{1}$$

where $\xi > 0$ is a weighting parameter between the accuracy of high-dimensional big data and the transmission rate of accurate data. a and d are model coefficients. $\mathrm{ade}(\cdot)$[34,35] is the choice function of r^i, $i \in \mathbf{N}^{+}$, which refers to the fact that the value of r^i, $i \in \mathbf{N}^{+}$, can meet the requirements of each end-users.

3.3 Problem formulation

QoE is one of the most crucial performance metrics for determining the quality of big data services in the pervasive edge computing environment. As well known that, many factors can affect QoE. In this study, we consider QoE from two perspectives: the accuracy of high-dimensional big data and the transmission rate of the accurate data. On one hand, we consider that the accuracy of high-dimensional big data can be determined based on whether the accurate data is positively proportional to the QoE. On the other hand, to meet the requirements of each end-user, in each time slot i, $i \in \mathbf{N}^{+}$, the transmission rate r^i, $i \in \mathbf{N}^{+}$, can not be less than the minimum $r^i_{\min}$, $i \in \mathbf{N}^{+}$, value or higher than the maximum $r^i_{\max}$, $i \in \mathbf{N}^{+}$, value of each end-user. In other words, we must realize transmission rate of r^i, $i \in \mathbf{N}^{+}$, to enhance the end-user QoE. We consider the above factors to enhance the quality of big data services in the pervasive edge computing environment. Finally, we employ the objective function ϕ, which represents the QoE and present the formulation of our QoE-maximization problem as follows:

$$\phi = \max \mathrm{QoE}(p^i, r^i) \sim (\max\ p^i) \cup \mathrm{ade}\,(r^i) \tag{2}$$

where $\sim$ is an equivalence symbol indicating that the value of QoE is equivalent to the value of p^i with the subscription of end-users in each time slot i, $i \in \mathbf{N}^{+}$.

4 Algorithm Design in the Pervasive Edge Computing Environment

Many methods have been used to manage the high-dimensional big data being generated in industry and academia, including big data analysis and data transmission rate management. However, deep learning[12,13] is a well-known dependable tool that is most often applied in big data analysis. Specifically, CNN, which is a branch of deep learning. In this section, to solve our second problem, we design a TF-CNN algorithm using tensor representation models[17] and truncated SVD[18] to extract accurate data while also improving the training speed. We also need a discrete method[36] for discretizing r^i, $i \in \mathbf{N}^{+}$, since the second term of the objective function is continuous according to the authors of a previous study addressing the QoE-maximization problem in the pervasive edge computing environment.

4.1 TF-CNN construction

More than any other methods, the TCNN[17] technique has been proposed as a way to improve training efficiency and ensure the accuracy of high-dimensional big data in the pervasive edge computing environment. Truncated SVD was introduced within CNN as a means for accelerating the training speed of high-dimensional big data analysis[18]. In fact, to complete the training process, the training of a TCNN with truncated SVD requires the accumulation of a sufficient number of samples by the algorithm proposed in Ref. [37]. Samples are deposited in an unsorted sum tree Z[38]. Based on these samples, we constructed the TF-CNN algorithm, the outline of which is provided by Algorithm 1.

4.2 TF-CNN training

First, we pretrain TF-CNN with Z using a high-order forward-pass[17] to obtain the output of each layer. Then, we compress the matrix Q_{IM} and Q_{FM}[18], both of them comprise the output of the layers. Lastly, the

Algorithm 1: Big Data Management based on TF-CNN

Input: Samples in Z, $threshold$, time i
Output: θ, r^i
Pretrain TF-CNN with Z
begin
  for *process in each time slot* i **do**
    Compute $q_{i1,\cdots,im}$ and $q_{f1,\cdots,fm}$;
    Compress $q_{i1,\cdots,im}$ and $q_{f1,\cdots,fm}$;
    if $U_{\text{TF-CNN}}(\theta) > threshold$ **then**
      if $t = m_t$ **then**
        $\varepsilon_f^{(t)} = \frac{\partial}{\partial q_f^{(t)}} = (g(q_f^{(t)}) - y_f) \cdot g'(q_f^{(t)})$;
      end
      for $t = 1, 2, \cdots, m_t - 1$ **do**
        if $typedeg = fullyconnected$ **then**
          $\varepsilon_{f1\cdots fm}^{(t)} = (D^{(t)})^T \odot \varepsilon_{i1\cdots im}^{(t+1)} \cdot g(q_{f1\cdots fm}^{(t)})$;
          $\bigtriangledown b^{(t)} = \sum \varepsilon_{i1\cdots im}^{(t+1)}$;
          $\bigtriangledown D^{(t)} = \sum a_{\alpha i1\cdots im}^{(t)} \varepsilon_{f1\cdots fm}^{(t+1)}$;
        end
        if $typedeg = pooling$ **then**
          $\varepsilon_{f1\cdots fm}^{(t)} = (B^{(t)})^T * \varepsilon_{i1\cdots im}^{(t)} \cdot g(q_{f1\cdots fm}^{(t)})$;
          $\bigtriangledown b^{(t)} = \sum \varepsilon_{i1\cdots im}^{(t+1)}$;
          $\bigtriangledown \beta^{(t)} = \sum \varepsilon_{i1\cdots im}^{(t+1)}$;
        end
        if $typedeg = convolution$ **then**
          $\varepsilon_{f1\cdots fm}^{(t)} = \beta^{(t+1)} \cdot (\text{up}(\varepsilon_{i1\cdots im}^{(t+1)}) \cdot g(q_{f1\cdots fm}^{(t)})$;
          $\bigtriangledown b^{(t)} = \sum \varepsilon_{i1\cdots im}^{(t+1)}$;
          $\bigtriangledown B^{(t)} = \sum a_{\alpha i1\cdots im}^{(t)} \cdot \varepsilon_{f1\cdots fm}^{(t+1)}$;
        end
      end
    end
    Update $\theta(D, b, B, \beta)$
  end
  Observe the accuracy $p^i(\theta)$ of data;
  for $r^i \in Q$, $i \in \mathbf{N}^+$ **do**
    discretize r^i, $i \in \mathbf{N}^+$;
  end
  Exe. $\phi = \max \text{QoE}(p^i, r^i)$.
end

TF-CNN is trained to perform fine-tuning by reducing a sequence of the loss function $U_{\text{TF-CNN}}$ to its minimum. Specifically, the loss function is denoted as follows:

$$U_{\text{TF-CNN}} = \frac{1}{2} \sum_{m=1}^{M} (s_\theta(x^m) - y^m) \tag{3}$$

where (x, y) is a tensor object, and s_θ refers to the dimensions.

To obtain the minimum $U_{\text{TF-CNN}}$, we first assign the weight of the tensors as random numbers. Then, the weight of the tensor is updated, and the implementation of the stochastic gradient method is as follows:

$$\begin{aligned} D_{f1\cdots fm}^{(t)} &= D_{f1\cdots fm}^{(t)} - \alpha \frac{\partial U_{\text{TF-CNN}}(\theta)}{\partial D_{f1\cdots fm}^{(t)}}, \\ b_{i1\cdots im}^{(t)} &= b_{i1\cdots im}^{(t)} - \alpha \frac{\partial U_{\text{TF-CNN}}(\theta)}{\partial b_{i1\cdots im}^{(t)}}, \\ B_{f1\cdots fm}^{(t)} &= B_{f1\cdots fm}^{(t)} - \alpha \frac{\partial U_{\text{TF-CNN}}(\theta)}{\partial B_{f1\cdots fm}^{(t)}}, \\ \beta_{i1\cdots im}^{(t)} &= \beta_{i1\cdots im}^{(t)} - \alpha \frac{\partial U_{\text{TF-CNN}}(\theta)}{\partial \beta_{i1\cdots im}^{(t)}} \end{aligned} \tag{4}$$

where $\frac{\partial U_{\text{TF-CNN}}(\theta)}{\partial k^{(t)}}$, $k = D, B, \beta, b$ represents the partial derivative, α is the learning rate, D is the $(M+1)$-order weight tensor, b is the bias tensor, B is M-order tensor, and β is the weight.

In the training process, the computation of each of the key steps is divided into three parts. First, the following sequence is computed using high-order forward pass:

$$\begin{aligned} q_{f1\cdots fm}^{(t)} &= D_{f1\cdots fm}^{(t-1)} \cdot X + b_{f1\cdots fm}^{(t-1)}, \\ q_{i1\cdots im}^{(t+1)} &= D_{i1\cdots im}^{(t)} \cdot g(z_{f1\cdots fm}^{(t)}) + b_{i1\cdots im}^{(t)} \end{aligned} \tag{5}$$

Then, the output layers are compressed using truncated SVD to accelerate the training speed[18]:

$$X \approx Q_{IM} \sum_l Q_{FM}^{\mathrm{T}} \tag{6}$$

where Q_{FM} is an $q_{fm} \times l$ matrix comprising the first l left-singular values of X, $\sum_l$ is an $l \times l$ diagonal matrix including the top l singular values of X, and Q_{IM} is $q_{im} \times l$ comprising the first l right singular matrix of X.

The last step is the computation of the partial derivatives during the updating process. The high-order partial derivatives are computed by performing efficient high-order backward propagation[17]. Then, the partial derivatives $\frac{\partial U_{\text{TF-CNN}}(\theta,x,y)}{\partial k^{(t)}}$, $k = D, B, \beta, b$ are implemented by the application of a tensor object (x, y) by the high-order backward propagation algorithm[17]. Lastly, the ultimate total partial derivatives $\frac{\partial U_{\text{TF-CNN}}(\theta,x,y)}{\partial k^{(t)}}$, $k = D, B, \beta, b$ are computed if those partial derivatives are obtained by the following:

$$\frac{\partial U_{\text{TF-CNN}}(\theta)}{\partial D^{(t)}_{f1\cdots fm}} = \frac{1}{M}\sum_{m=1}^{M}\frac{\partial U_{\text{TF-CNN}}(\theta, x^{(i)}, y^{(i)})}{\partial D^{(t)}_{f1\cdots fm}},$$
$$\frac{\partial U_{\text{TF-CNN}}(\theta)}{\partial \beta^{(t)}_{i1\cdots im}} = \frac{1}{M}\sum_{m=1}^{M}\frac{\partial U_{\text{TF-CNN}}(\theta, x^{(i)}, y^{(i)})}{\partial \beta^{(t)}_{i1\cdots im}},$$
$$\frac{\partial U_{\text{TF-CNN}}(\theta)}{\partial b^{(t)}_{i1\cdots im}} = \frac{1}{M}\sum_{m=1}^{M}\frac{\partial U_{\text{TF-CNN}}(\theta, x^{(i)}, y^{(i)})}{\partial b^{(t)}_{i1\cdots im}},$$
$$\frac{\partial U_{\text{TF-CNN}}(\theta)}{\partial B^{(t)}_{f1\cdots fm}} = \frac{1}{M}\sum_{m=1}^{M}\frac{\partial U_{\text{TF-CNN}}(\theta, x^{(i)}, y^{(i)})}{\partial B^{(t)}_{f1\cdots fm}} \tag{7}$$

4.3 Discrete transmission rate

After training high-dimensional big data using TF-CNN, we can obtain accurate data. Then, we discretize the transmission rate r^i, $i \in \mathbf{N}^+$, of the accurate data to meet the requirements of each end-users.

To solve our second problem, we propose a method, shown in Fig. 2 for discretizing r^i, $i \in \mathbf{N}^+$, since the second term of the objective function is continuous as reported by the authors of previous studies[39–41]. Because the range of r^i, $i \in \mathbf{N}^+$, is $[r^i_{\min}; r^i_{\max}]$, we divide it into h intervals, each of which has the same size H, and use the h division points as the discrete values of r^i, $i \in \mathbf{N}^+$. We include all the division points in set Q. We can see that when $h \to \infty$, $H \to 0$, the values of r^i, $i \in \mathbf{N}^+$, are close to continuous. For example, Q_4, $i = 4$, consists of five elements, including $r_{\min}$, $r_{\max}$ and three division points. $Q_4 = \{r_{\min}; r_{\min} + (r_{\max} - r_{\min})/4; r_{\min} + 2(r_{\max} - r_{\min})/4; r_{\min} + 3(r_{\max} - r_{\min})/4; r_{\max}\}$.

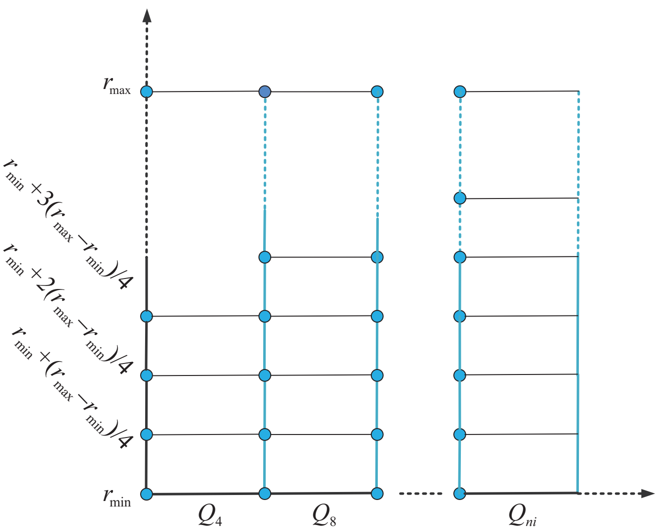

Fig. 2 The $r_{\min}$ and $r_{\max}$ components belong to every set Q, Q includes the i division point.

In general, the management process for Algorithm 1 includes a training process for the high-dimensional big data and a discretization process of the transmission rate, which are executed as follows. Note that, $\bigtriangledown$ refers to $\frac{\partial U_{\text{TF-CNN}}(\theta,x,y)}{\partial k^{(t)}}$, $k = D, B, \beta, b$ in Algorithm 1.

Step 1. Execute a forward pass and compute the outputs $q_{i1\cdots im}$, $q_{f1\cdots fm}$ of each layer (line 6).

Step 2. Compress the $q_{i1\cdots im}, q_{f1\cdots fm}$ values of the output layer (line 7).

Step 3. Compute the error term $\varepsilon^{(t)}_f$ by performing backward propagation for every unit f in the output layer as follows (lines 9–11):

$$\varepsilon^{(t)}_f = \frac{\partial U_{\text{TF-CNN}}(\theta)}{\partial q^{(t)}_f} = (g(q^{(t)}_f) - y_f) \cdot g'(q^{(t)}_f) \tag{8}$$

where z_j is the element of tensor X.

Step 4. Compute the error term $\varepsilon^{(t)}_f$ as well, by performing backward propagation for every unit f in the layer $t = m_t - 1, \ldots, 3, 2$.

- **Layer t in a tensor fully connected layer** (lines 13 and 14):

$$\varepsilon^{(t)}_{f1\cdots fm} = \frac{\partial U_{\text{TF-CNN}}(\theta, x, y)}{\partial q^{(t)}_{f1\cdots fm}} = \sum_{i1=1}^{m}\cdots\sum_{im=1}^{m} w^{(t)}_{i1\cdots im f1\cdots fm}\varepsilon^{(t+1)}_{i1\cdots im} \cdot g(q^{(t)}_{f1\cdots fm}) = (D^{(t)})^{\mathrm{T}} \odot \varepsilon^{(t+1)}_{i1\cdots im} \cdot g(q^{(t)}_{f1\cdots fm}) \tag{9}$$

where $w^{(t)}_{i1\ldots im f1\ldots fm}$ denotes the weight difference between unit $f1\ldots fm$ of layer t and unit $i1\ldots im$ of layer $t+1$.

- **Layer t in a tensor pooling layer** (lines 18 and 19):

$$\varepsilon^{(t)}_{f1\cdots fm} = \frac{\partial U_{\text{TF-CNN}}(\theta, x, y)}{\partial q^{(t)}_{f1\cdots fm}} = \sum_{i1=1}^{m}\cdots\sum_{im=1}^{m} k^{(t)}_{f1\cdots fm}\varepsilon^{(t+1)}_{i1\cdots im} \cdot g(q^{(t)}_{f1\cdots fm}) = (B^{(t)})^{\mathrm{T}} \cdot \varepsilon^{(t+1)}_{i1\cdots im} \cdot g(q^{(t)}_{f1\cdots fm}) \tag{10}$$

where $k^{(t)}_{f1\cdots fm}$ is the weight of kernel L.

- **Layer t in a tensor convolutional layer** (lines 23 and 24):

$$\varepsilon^{(t)}_{f1\cdots fm} = \frac{\partial U_{\text{TF-CNN}}(\theta, x, y)}{\partial q^{(t)}_{f1\cdots fm}} = \beta^{(t+1)}(\mathrm{up}(\varepsilon^{(t+1)}_{i1\cdots im})) \cdot g(q^{(t)}_{f1\cdots fm}) \tag{11}$$

where $\mathrm{up}(\varepsilon_{i1\cdots im}^{(t+1)})$ uses the factor sc^i to tile the input element of each dimension, which is referred to as an upsampling operation.

Step 5. Compute the error term $\frac{\partial q^{(t+1)}}{\partial k^{(t)}}, k = D, B$ as follows (lines 16 and 26):

$$\frac{\partial q_{i1\cdots im}^{(t+1)}}{\partial k_{\alpha i1\cdots im}^{(t)}} = a_{\alpha i1\cdots im}^{t} \tag{12}$$

Step 6. Compute the desired partial derivatives as well, as follows (lines 15, 16, 20, 21, 25, 26):

$$\begin{aligned}
\frac{\partial U_{\mathrm{TF-CNN}}(\theta, x, y)}{\partial D_{f1\cdots fm}^{(t)}} &= \sum a_{\alpha i1\cdots im}^{(t)} \cdot \varepsilon_{f1\cdots fm}^{(t+1)},\\
\frac{\partial U_{\mathrm{TF-CNN}}(\theta, x, y)}{\partial B_{f1\cdots fm}^{(t)}} &= \sum a_{\alpha i1\cdots im}^{(t)} \cdot \varepsilon_{f1\cdots fm}^{(t+1)},\\
\frac{\partial U_{\mathrm{TF-CNN}}(\theta, x, y)}{\partial b_{i1\cdots im}^{(t)}} &= \sum \varepsilon_{i1\cdots im}^{(t+1)},\\
\frac{\partial U_{\mathrm{TF-CNN}}(\theta, x, y)}{\partial \beta_{i1\cdots im}^{(t)}} &= \sum \varepsilon_{i1\cdots im}^{(t+1)}
\end{aligned} \tag{13}$$

Step 7. The TF-CNN updates the parameters D, b, B, β in each time i, to obtain the accuracy $p^i(D, b, B, \beta)$ of the high-dimensional big data (line 32).

Step 8. Discretize the transmission rate $r^i, i \in \mathbf{N}^+$, of accurate data (lines 33–35).

Step 9. Obtain ϕ by p^i and $r^i, i \in \mathbf{N}^+$ (line 36).

4.4 Algorithm complexity

In an FR-CNN algorithm[18], the complexity is $O(PLs^2)$ in a convolutional layer with P output mapping the $R^{s\times s}$ and L kernels, and the complexity is $O(Ss^2)$ in a pooling layer with S outputs $R^{s\times s}$, the complexity is $O(mn)$ in a fully connected layer with the weight $D^{m\times n}$. Thus, the total complexity of FR-CNN is less than $O(PLs^2 + Ss^2 + mn)$. Similarly, the TCNN[17] is the generalized algorithm for FR-CNN, which extends CNN from the vector space to the tensor space using an accelerator. Therefore, the complexity of TCNN is $O(PLs^M + Ss^M + Q_{IM}Q_{FM+1})$, $Q_{IM} = Q_{I1} \times Q_{I2} \times \cdots \times Q_{IM}$ and $Q_{FM+1} = Q_{F1} \times Q_{F2} \times \cdots \times Q_{FM+1}$ represent the orders of input and weight. Compared to the TCNN algorithm, the complexity of our proposed TF-CNN is reduced by half. However, the complexity of the TF-CNN is slightly higher than that of FR-CNN. Similarly, the complexity of big data management based on the TF-CNN algorithm is a little higher than that of TF-CNN.

5 Performance Evaluations

The quality of high-dimensional big data services is chiefly determined by the data accuracy and the transmission rate of accurate data. Therefore, we selected advanced algorithms to analyze our collections of big data in the pervasive edge computing environment to extract the most accurate data. Then, we manage the transmission rate of the accurate data to meet the requirements of each end-users. In this section, we quantitatively evaluate and validate the performance of our proposed algorithm for high-dimensional big data management.

5.1 Experiment settings

We conducted our evaluation via a two-step experiment. In the first step, we implemented the TCNN, FR-CNN, and TF-CNN algorithms, in MATLAB[42], and evaluated them using a TensorFlow[43] simulation tool written in Python, using the parameters listed in Table 2. Note that, TF-CNN is our proposed scheme, and TCNN and FR-CNN are drawn from other works. We employed this commonly used simulator because it is designed to import a realistic trace as input from all types of database. We also used the SNAE2 data set[44], which is collected by sensors from different sources, and in various raw-data formats in the pervasive edge computing environment. We chose 2 TB data in the simulation and also explored the important factors that influence the QoE. Our goal was to optimize the data transmission. Finally, we also used TensorFlow to evaluate the performance of the QoE models.

In the second step, using a specific video-streaming service, we conducted a comparative experiment of the proposed algorithms using different transmission rates to determine their effect on QoE, as computed by the aforementioned QoE model. Generally, different management methods are based on the quality of video chosen by end-users. Then, management methods are employed to measure end-user QoE.

Table 2 Simulation parameters.

Parameter	Value
Hidden layer 1	256 neurons
Hidden layer 2	64 neurons
Hidden layer 3	32 neurons
Hidden layer 4	8 neurons
Discount parameter	0.8
Learning rate	0.2

5.2 Results

For each test case, we used TensorFlow to evaluate the performances and efficiencies of the three algorithms in terms of their accuracy, precision, and recall. We note that, the accuracy is defined as the ratio of the number of samples divided by the total number of samples correctly sorted by the classifier for a given test data set. The Precision is defined as the ratio between the number of corresponding documents retrieved, and the total number of documents as measured by the search system. The Recall refers to the ratio of the number of documents with respect to the number of corresponding documents in the document library. In Fig. 3, we can see that the performance of FR-CNN in terms of accuracy, precision, and recall is the worst, whereas that of TF-CNN is the best. Next, we evaluated the performance of the algorithms in terms of training time on the data set. As shown in Fig. 4, due to the SVD, TF-CNN takes less times to analyze the data than TCNN, although the training time is longer than that of FR-CNN. This is why we adopted the TF-CNN algorithm to analyze high-dimensional big data. We also validated the QoE factor while varying the transmission rate with h. We divided r^i, $i \in \mathbf{N}^+$, into h intervals and set the value of the h division points as Q. When h becomes larger, the number of division points increases.

Figures 5 and 6 show the results, from which we can see that the average accuracies of FR-CNN, TCNN, and TF-CNN were 0.83, 0.85, and 0.87, respectively. We also find that, the performance accuracy p^i, $i \in \mathbf{N}^+$, of FR-CNN is the worst, whereas TF-CNN is the best. We can also see that the QoE values change greatly as p^i, $i \in \mathbf{N}^+$, increases in Fig. 6. In this figure, we can see that the TF-CNN algorithm generates the highest QoE value, and the FR-CNN algorithm the lowest. Clearly, when k increases, QoE increases. However, as we can see in Fig. 7, when the transmission rate becomes faster than normal, the QoE value remains nearly the same or even decreases. Previous studies[45,46] have reported that a satisfactory QoE value is usually above 60. Therefore, we can conclude that a high-dimensional big data management method based on the TF-CNN algorithm, which exhibits the best accuracy p^i and r^i, $i \in \mathbf{N}^+$, provides end-users with a satisfactory QoE.

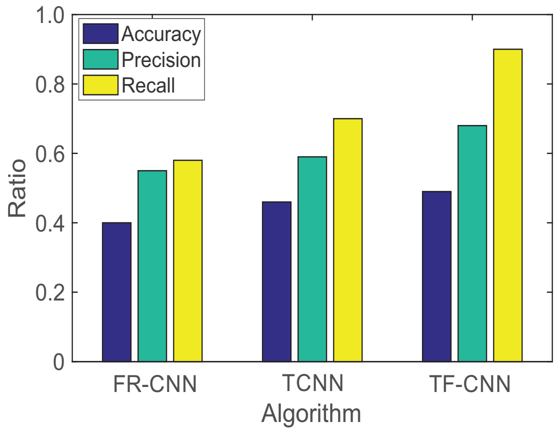

Fig. 3 Performance with different algorithms.

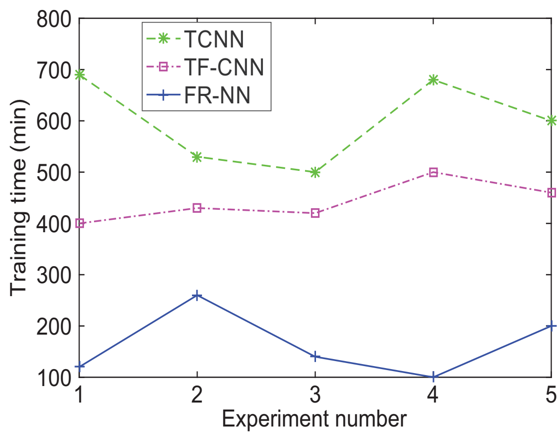

Fig. 4 Training time with different algorithms.

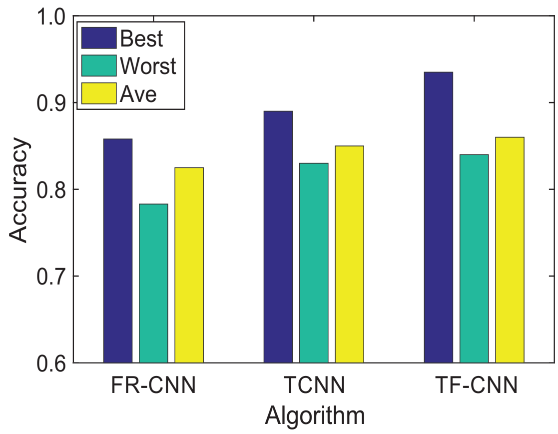

Fig. 5 Accuracy for different algorithms.

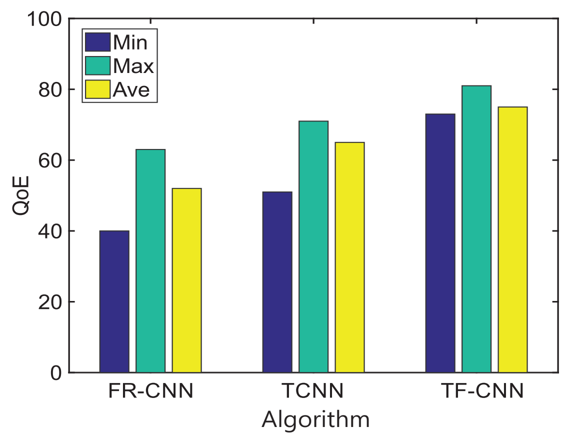

Fig. 6 QoE values for different algorithms.

From Figs. 3–6, we can see that the QoE performances of the three algorithms are positively correlated to the efficiency of their performance in terms of accuracy.

From Fig. 7, we find that the QoE performances are positively related to adequate transmission rates. Therefore, we can realize better performances by QoE models shown in Fig. 8 because the QoE model with r^i and TF-CNN perform better than other QoE models. From our above discussion and from Fig. 9, we can see that QoE performances are positively related to the accuracy of high-dimensional big data as well as the adequacy of the transmission rate of accurate data.

6 Conclusion

In the work, we investigated high-dimensional big data management in the pervasive edge computing environment and methods for improving the QoE of end-users. We also analyzed the factors that impact QoE. To obtain accuracy in high-dimensional big data as quickly as possible, we proposed the TF-CNN algorithm and a high-dimensional big data management method based on TF-CNN. Our experimental results revealed the high QoE performance of our proposed high-dimensional big data management method.

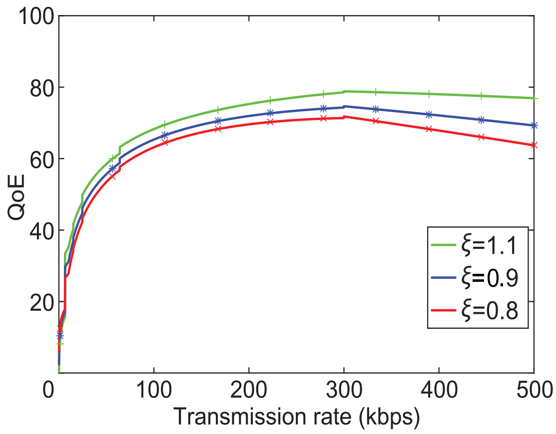

Fig. 7 QoE performance for various transmission rates via *h*, for different ξ.

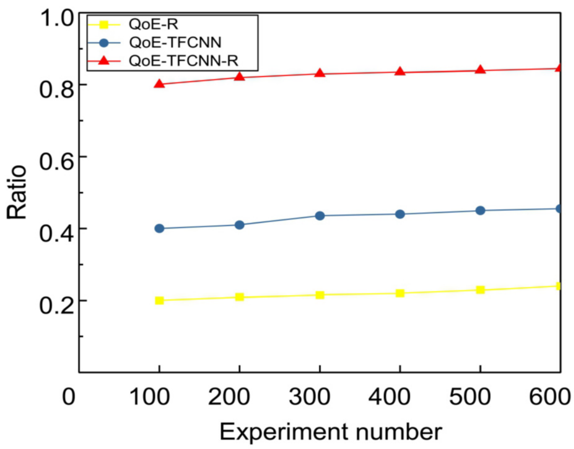

Fig. 8 Performance of different QoE models.

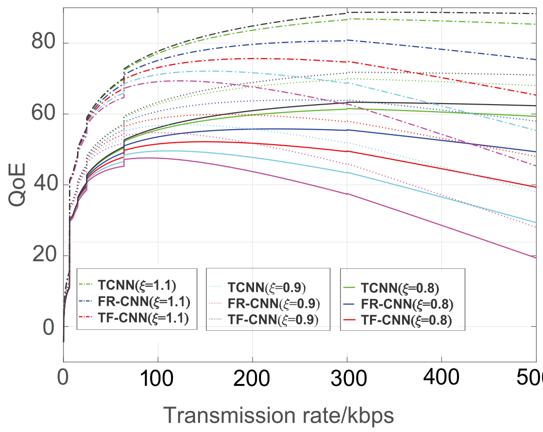

Fig. 9 QoE performance under different algorithms v.s. varying transmission rate, under different ξ.

Acknowledgment

This work was supported by the National Key Basic Research and Development (973) Program of China (No. 2015CB352401), the National Natural Science Foundation of China (Nos. 61572262 and 61772286), and China Postdoctoral Science Foundation (No. 2017M610252), China Postdoctoral Science Special Foundation (No. 2017T100297).

References

[1] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, Green industrial Internet of Things architecture: An energy-efficient perspective, *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 48–54, 2016.

[2] M. Guo, E. Olule, G. Wang, and S. Guo, Designing energy efficient target tracking protocol with quality monitoring in wireless sensor networks, *Journal of Supercomputing*, vol. 51, no. 2, pp. 131–148, 2010.

[3] Q. Meng, K. Wang, B. Liu, T. Miyazaki, and X. He, QoE-based big data analysis with deep learning in pervasive edge environment, in *Proc. Int. IEEE Communications Conf.*, Kansas City, MO, USA, 2018.

[4] X. Song, Y. Huang, Q. Zhou, F. Ye, Y. Yang, and X. Li, Pervasive edge data sharing in MANET, in *Proc. Int. IEEE Computer Communications Workshops Conf.*, Atlanta, GA, USA, 2017, pp. 133–138.

[5] C. Xu, J. Ren, Y. Zhang, Z. Qin, and K. Ren, DPPro: Differentially private high-dimensional data release via random projection, *IEEE Trans. Information Forensics and Security*, vol. 12, no. 12, pp. 3081–3093, 2017.

[6] B. Wang and K. Mueller, The subspace voyager: Exploring high-dimensional data along a continuum of salient 3D subspaces, *IEEE Trans. Visualization and Computer Graphics*, vol. 24, no. 12, pp. 1204–1222, 2018.

[7] Y. Chen, K. Wu, and Q. Zhang, From QoS to QoE: A tutorial on video quality assessment, *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 2, pp. 1126–1165, 2014.

[8] X. Zhou, K. Wang, W. Jia, and M. Guo, Reinforcement learning-based adaptive resource management of differentiated services in geo-distributed data centers, in *Proc. Int. IEEE/ACM Symp. Quality of Service Conf.*, Vilanova, Spain, 2017, pp. 1–6.

[9] Z. Ye, S. Mistry, A. Bouguettaya, and H. Dong, Long-term QoS-aware cloud service composition using multivariate time series analysis, *IEEE Trans. Services Computing*, vol. 9, no. 3, pp. 382–393, 2016.

[10] S. Bulo, B. Biggio, I. Pillai, M. Pelillo, and F. Roli, Randomized prediction games for adversarial machine learning, *IEEE Trans. Neural Networks and Learning Systems*, vol. 28, no. 11, pp. 2466–2478, 2017.

[11] M. Li, J. Wei, X. Zheng, and M. Bolton, A formal machine-learning approach to generating human-machine interfaces from task models, *IEEE Trans. Human-Machine Systems*, vol. 47, no. 6, pp. 822–833, 2017.

[12] H. Wu and S. Prasad, Semi-supervised deep learning using pseudo labels for hyperspectral image classification, *IEEE Trans. Image Processing*, vol. 27, no. 3, pp. 1259–1270, 2017.

[13] Z. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems, *IEEE Commun. Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[14] M. Federico, P. Julian, and P. Mandolesi, SCDVP: A simplicial CNN digital visual processor, *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 61, no. 7, pp. 1962–1969, 2014.

[15] C. Hsu and C. Lin, CNN-based joint clustering and representation learning with feature drift compensation for large-scale image data, *IEEE Trans. Multimedia*, vol. 20, no. 2, pp. 421–429, 2017.

[16] H. Lee, K. Hong, H. Kang, and S. Lee, Photo aesthetics analysis via DCNN feature encoding, *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1921–1932, 2017.

[17] P. Li, Z. Chen, L. Yang, Q. Zhang, and M. Deen, Deep convolutional computation model for feature learning on big data in internet of things, *IEEE Trans. Industrial Informatics*, DOI: 10.1109/TII.2017.2739340.

[18] R. Girshick, Fast R-CNN, in *Proc. Int. IEEE Computer Vision Conf.*, Santiago, Chile, 2015, pp. 1440–1448.

[19] Q. Chen, M. Guo, Q. Deng, L. Zheng, S. Guo, and Y. Shen, HAT: History-based auto-tuning MapReduce in heterogeneous environments, *Journal of Supercomputing*, vol. 64, no. 3, pp. 1038–1054, 2011.

[20] T. Zhao, Q. Liu, and C. Chen, QoE in video transmission: A user experience-driven strategy, *IEEE Commun. Surveys & Tutorials*, vol. 19, no. 1, pp. 285–302, 2016.

[21] S. Kim, G. Suk, J. Lee, and C. Chae, QoE-aware scalable video transmission in MIMO systems, *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 196–203, 2017.

[22] C. Liang, Y. He, F. Yu, and N. Zhao, Enhancing QoE-aware wireless edge caching with software-defined wireless networks, *IEEE Trans. Wireless Communications*, vol. 16, no. 10, pp. 6912–6925, 2017.

[23] K. Wang, L. Gu, S. Guo, H. Chen, V. Leung, and Y. Sun, Crowdsourcing-based content-centric network: A social perspective, *IEEE Network*, vol. 31, no. 5, pp. 28–34, 2017.

[24] C. Ji, T. Dong, Y. Li, Y. Shen, K. Li, W. Qiu, W. Qu, and M. Guo, Inverted grid-based KNN query processing with MapReduce, in *Proc. Int. IEEE ChinaGrid Annual Conf.*, Beijing, China, 2012, pp. 25–32.

[25] S. Zhao and D. Medhi, Application-aware network design for Hadoop MapReduce optimization using software-defined networking, *IEEE Trans. Network and Service Management*, vol. 14, no. 4, pp. 804–816, 2017.

[26] W. Shi, Y. Gong, X. Tao, J. Wang, and N. Zheng, Improving CNN performance accuracies with min-max objective, *IEEE Trans. Neural Networks and Learning Systems*, DOI: 10.1109/TNNLS.2017.2705682.

[27] Z. Zhang, T. Weng, and L. Daniel, Big-data tensor recovery for high-dimensional uncertainty quantification of process variations, *IEEE Trans. Components, Packaging and Manufacturing*, vol. 7, no. 5, pp. 687–697, 2017.

[28] M. Li and X. Wang, Delay and rate satisfaction for data transmission with application in wireless communications, *IEEE Network*, vol. 29, no. 5, pp. 70–75, 2015.

[29] R. Borujeny, M. Noori, and M. Ardakani, Maximizing data rate for multiway relay channels with pairwise transmission strategy, *IEEE Trans. Wireless Communications*, vol. 16, no. 3, pp. 1609–1618, 2017.

[30] K. Wang, X. Qi, L. Shu, D. Deng, and J. Rodrigues, Toward trustworthy crowdsourcing in social internet of things, *IEEE Wireless Communications*, vol. 30, no. 5, pp. 30–36, 2016.

[31] D. Wu, B. He, X. Tang, J. Xu, and M. Guo, RAMZzz: Rank-aware DRAM power management with dynamic migrations and demotions, in *Proc. Int. IEEE High Performance Computing, Networking, Storage and Analysis Conf.*, Salt Lake City, UT, USA, 2012, pp. 1–11.

[32] C. Xu, K. Wang, and M. Guo, Intelligent resource management in blockchain based cloud data centers, *IEEE Cloud Computing*, vol. 4, no. 6, pp. 50–59, 2017.

[33] Q. Chen, Y. Chen, Z. Huang, and M. Guo, WATS: Workload-aware task scheduling in asymmetric multi-core architectures, in *Proc. Int. 26th IEEE Parallel and Distributed Processing Symposium Conf.*, Shanghai, China, 2012, pp. 249–260.

[34] Z. Mei, Y. Zhang, X. Zhao, B. Jung, T. Sarkar, and M. Palma, Choice of the scaling factor in a marching-on-in-degree time domain technique based on the associated laguerre functions, *IEEE Trans. Antennas and Propagation*, vol. 60, no. 9, pp. 4463–4467, 2012.

[35] H. Huang, P. Li, S. Guo, W. Liang, and K. Wang, Near-optimal deployment of service chains by exploiting correlations between network functions, *IEEE Trans. Cloud Computing*, DOI:10.1109/TCC.2017.2780165.

[36] G. Silva, R. Vieira, and C. Rech, Discrete-time sliding-mode observer for capacitor voltage control in modular

multilevel converters, *IEEE Trans. Industrial Electronics*, vol. 65, no. 1, pp. 876–886, 2018.

[37] R. Girshick, Photo aesthetics analysis via DCNN feature encoding, in *Proc. Int. IEEE Computer Vision Conf.*, Santiago, Chile, 2015, pp. 1921–1932.

[38] Y. Xu, J. Chen, and Q. Wang, The sum rate of vector gaussian multiple description coding with tree-structured covariance distortion constraints, *IEEE Trans. Information Theory*, vol. 63, no. 10, pp. 6747–6560, 2017.

[39] K. Wang, J. Mi, C. Xu, Q. Zhu, L. Shu, and D. Deng, Real-time load reduction in multimedia big data for mobile Internet, *ACM Trans. Multimedia Computing, Communications and Applications*, vol. 12, no. 5s, p. 76, 2016.

[40] K. Wang, H. Gao, X. Xu, J. Jiang, and D. Yue, An energy-efficient reliable data transmission scheme for complex environmental monitoring in underwater acoustic sensor networks, *IEEE Sensors Journal*, vol. 16, no. 11, pp. 4051–4062, 2016.

[41] K. Wang, Y. Shao, L. Shu, Y. Zhang, and C. Zhu, Mobile big data fault-tolerant processing for eHealth networks, *IEEE Network*, vol. 30, no. 1, pp. 1–7, 2016.

[42] Z. Wang, J. Yang, R. Melhem, B. Childers, Y. Zhang, and M. Guo, Simultaneous multikernel GPU: Multi-tasking throughput processors via fine-grained sharing, in *Proc. Int. IEEE Symp. High Performance Computer Architecture Conf.*, Barcelona, Spain, 2016, pp. 358–369.

[43] X. He, K. Wang, T. Miyazaki, H. Huang, Y. Wang, and S. Guo, Green resource allocation based on deep reinforcement learning in content-centric IoT, *IEEE Trans. Emerging Topics in Computing*, DOI:10.1109/TETC.2018.2805718.

[44] Z. Li, Y. Shen, B. Yao, and M. Guo, OFScheduler: A dynamic network optimizer for MapReduce in heterogeneous cluster, *International Journal of Parallel Programming*, vol. 43, no. 3, pp. 472–488, 2013.

[45] 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. 2–8, 2018.

[46] W. Rahman, D. Yun, and K. Chung, A client side buffer management algorithm to improve QoE, *IEEE Trans. Consumer Electronics*, vol. 62, no. 4, pp. 371–379, 2016.

Qianyu Meng is a research member in Jiangsu Engineering Research Center of Communication and Network Technology, Nanjing University of Posts and Telecommunications. She is also a master student in College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. She received the BEng degree from Nanjing University of Information Science and Technology, China, in 2012. Her current research interests include big data, machine learning, pervasive computing.

Kun Wang received the BEng and PhD degrees in computer science from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004 and 2009, respectively. From 2013 to 2015, he was a postdoc fellow in Electrical Engineering Department, University of California, Los Angeles (UCLA), CA, USA. In 2016, he was a research fellow in the School of Computer Science and Engineering, the University of Aizu, Aizu-Wakamatsu City, Fukushima, Japan. He is currently a research fellow in the Department of Computing, the Hong Kong Polytechnic University, Hong Kong, China, and also a full professor in the School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China. He has published over 100 papers in referred international conferences and journals. He has received Best Paper Award at IEEE GLOBECOM16. He serves as an associate editor of *IEEE Access*, editor of *Journal of Network and Computer Applications*, *Journal of Communications and Information Networks*, *EAI Transactions on Industrial Networks and Intelligent Systems*, and guest editors of *IEEE Access*, *Future Generation Computer Systems*, *Peer-to-Peer Networking and Applications*, and *Journal of Internet Technology*. He was the symposium chair/co-chair of IEEE IECON16, IEEE EEEIC16, IEEE WCSP16, IEEE CNCC17, etc. His current research interests are mainly in the area of big data, wireless communications and networking, smart grid, energy Internet, and information security technologies. He is a senior member of IEEE and member of ACM.

Xiaoming He is a master student in Jiangsu Engineering Research Center of Communication and Network Technology, Nanjing University of Posts and Telecommunications. He received the BEng degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2012. His current research interests include vehicle-to-grid, big data, machine learning, content-centric network, and pervasive computing.

Minyi Guo received the PhD degree in computer science from the University of Tsukuba, Tsukuba, Japan. He is currently a Zhiyuan Chair Professor in Shanghai Jiao Tong University, Shanghai, China. His research interests include pervasive computing, parallel and distributed processing, and parallelizing compilers. In 2007, he received the Recruitment Program of Global Experts and Distinguished Young Scholars Award from the National Natural Science Foundation of China. He is on the editorial board of *IEEE Transactions on Parallel and Distributed Systems*.