Redundancy Avoidance for Big Data in Data Centers: A Conventional Neural Network Approach

Chenhan Xu, Kun Wang, *Senior Member, IEEE*, Yanfei Sun, Song Guo, *Senior Member, IEEE*, and Albert Y. Zomaya, *Fellow, IEEE*

Abstract—As the innovative data collection technologies are applying to every aspect of our society, the data volume is skyrocketing. Such phenomenon poses tremendous challenges to data centers with respect to enabling storage. In this paper, a hybrid-stream big data analytics model is proposed to perform multimedia big data analysis. This model contains four procedures, i.e., data pre-processing, data classification, data recognition and data load reduction. Specifically, an innovative multi-dimensional Convolution Neural Network (CNN) is proposed to assess the importance of each video frame. Thus, those unimportant frames can be dropped by a reliable decision-making algorithm. In order to ensure video quality, minimal correlation and minimal redundancy (MCMR) are combined to optimize the decision-making algorithm. Simulation results show that the amount of processed video is significantly reduced, and the quality of video is preserved due to the addition of MCMR. The simulation also proves that the proposed model performs steadily and is robust enough to scale up to accommodate the big data crush in data centers.

Index Terms—Data centers, redundancy avoidance, multimedia, storage, big data, convolution neural network

1 Introduction

Accelerating content delivery across the data centers [1], [2] has been a key enabler for myriad applications. The emergence of modern techniques for data generation and data collection has posed further challenges in transmitting and storing big data in a timely fashion. Transmitting and storing big data, especially, videos and images, within the realm of data centers is indeed very challenging [3], [4], [5], [6]. On the one hand, higher resolution is required to ensure the fidelity and granularity of the multimedia data; on the other hand, current storage architectures have not kept up with the influx of such big data [7], [8], [9], [10], [11]. Meanwhile, Liu et al. [12] showed that the disk utilization is not really high. Since it is desirable to provision content with superior quality and transmission with sufficient bandwidth, a method capable of reducing the load of the storage system is now in desperate need.

- *C. Xu and Y. Sun are with the Jiangsu Engineering Research Center of Communication and Network Technology and National Engineering Research Center of Communications and Networking, Nanjing University of Posts and Telecommunications, Nanjing 210003, China. E-mail: xchank@outlook.com, sunyanfei@njupt.edu.cn.*
- *K. Wang and S. Guo are with Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail: {cskun.wang, song.guo}@polyu.edu.hk.*
- *A. Zomaya is with School of Information Technologies, The University of Sydney, Sydney, NSW 2006, Australia. E-mail: albert.zomaya@sydney.edu.au.*

Manuscript received 21 Dec. 2017; revised 17 May 2018; accepted 29 May 2018. Date of publication 4 June 2018; date of current version 5 Mar. 2020. (Corresponding author: Kun Wang.)
Recommended for acceptance by J. Li, J. Wu, B. Hu, C. Wang, M. Daneshmand, and R. Malekian.
Digital Object Identifier no. 10.1109/TNSE.2018.2843326

A strategy that could reduce the transmission load among data centers and leverage the storage system is required to address the above problems. Some studies on reducing the transmission load have focused on optimizing route selection as well as detecting and dropping anomaly traffic. Vertical handoff (VHO) decision algorithm performs well in heterogeneous wireless networks [13], but when the time dimension is considered in multimedia transmission and storage, VHO becomes extremely complex. Convolution Neural Network, which incorporates pooling, can improve generalization on pattern recognition problems by sharing weights and biases. Hybrid Convolutional Neural Network (HCNN) combines CNN and winner-takes-all mechanism to further boost the recognition speed [14]. A two-stream CNN structure [15] and its updated version [16] have been proposed to process and classify videos by categorizing the original video information into spatial and temporal information. These two models have enhanced the performance of CNN, but still fall short of achieving real-time processing. Researches on making full use of storage systems principally focus on evaluating server load and selecting the best node for performing storage task. Liu et al. [12] proposed a strategy for load balancing of virtual storage that selects the best storage node by the calculated weight, and Prabavathy et al. [17] proposed an approach to balancing the load during the data placement as well as in any load imbalance situations, but they both paid no attention to the particularity of videos. Zhou et al. [18] proposed a black-box method to deal with load measurement and dynamic load balance. However, they used a static threshold depending on the actual system requirements that cannot respond well to the influx of data.

This paper addresses the issue of data redundancy, with emphasis on the big multimedia data in data centers. Inspired from the two-stream model that separates input video information into spatial and temporal information, a hybrid-stream big data analytics model is proposed in this paper. The static information (such as scenes) and dynamic information (such as gestures) are processed by two different conventional layers. Consider a sequence of frames or clips that makes up a video stream, and each frame in the sequence is an image. Then, to mitigate multimedia data redundancy in data centers, we can distinguish the abnormal traffic from normal data centers' traffic. To this end, a hybrid-stream big data analytics model, which consists of a dependable key frame extraction mechanism and a multi-dimensional CNN classification algorithm, is proposed to improve the classification precision and reduce the multimedia data redundancy in data centers.

Nevertheless, we are facing several challenges in facilitating multimedia big data analysis. The primary challenge is how to enhance the precision of video classification. Generally, a majority of video classification algorithms attempt to reduce the video dimension and use low dimensional representations [19]. In our model, a multi-dimensional CNN algorithm with a time dimension is proposed. As compared with classical CNN algorithms, this proposed algorithm is able to enhance the classification precision. Besides, how to process the classified outcomes with reduced multimedia data redundancy is another key issue. This is accomplished by incorporating a key frame extraction mechanism in our model. With this mechanism, this paper focuses on finding a proper proportion of reduced redundancy data, which reduces important data loss. Specifically, four main steps are presented, including Video Pre-processing, Frame Classification, Frame-Load-Reduction Processing, and Video Decision. Pre-processing provides an intermediate layer to adapt input video streams to our model. The purpose of Frame Classification is to evaluate the significance of each frame. Simultaneously, Frame-Load-Reduction Processing and Video Decision are proposed to perform data redundancy avoidance and ensure vital videos are not dropped. Generally, our work focuses on enabling real-time video processing by enhancing the storage efficiency in terms of storing useful data, and by further relieving multimedia data redundancy in data centers.

The contributions of our paper are summarized as follows.

- A hybrid-stream big data analytics model aimed to solve the multimedia big data redundancy problem in data centers is proposed. Different types of inputs are used to improve the classification accuracy. The experimental results show that this model can reduce the multimedia big data redundancy by about 4 percent, which means a huge amount of computational resources can be saved.
- A multi-dimensional CNN classification algorithm used to recognize video information is presented. By taking the time dimension into account, the convolution in CNN can further reduce the computational complexity. The time dimension is also involved in the pooling operation, which provides extra information for classifying video frames and clips more accurately. We are the first attempt to use multi-dimensional video information for reducing the storage load in data centers.
- A key frame extraction mechanism is proposed to realize load-reduction. Furthermore, we propose minimal correlation and minimal redundancy (MCMR) which can cover the shortage of key frame extraction. It is able to reserve the important frames which are dropped by key frame extraction mistakenly. To optimize the objective quality of videos after load-reduction, a special threshold is designed to control the MCMR, and Genetic Algorithm (GA) is used to determine this threshold.

The rest of this paper is organized as follows. Our system model is illustrated in the next section. Sections 3 and 4 cover the classification module and the load-reduction module in detail. In Section 5, we illustrate the system performance by different types of comparisons. Section 6 summaries the related work, and Section 7 draws a conclusion of this paper.

2 System Model

In data centers, video data may surge at any time. However, the frequent transmission and storage of the same content may exhaust the processing capability of data centers. Considering the capacity of the transmission load, the redundancy of multimedia data in data centers cannot be ignored [20]. However, when the data are transferred among data centers, one has to consider how to efficiently store these data [20], [21]. We propose an efficient algorithm to classify these data and then store them by the results of the classification. In other words, the key of our proposed algorithm is to reduce the redundancy of data and store them as soon as possible.

Generally, a video sequence is made up of numerous frames. In order to reduce the multimedia data redundancy and ensure that the data can be stored quickly and precisely, a mechanism is proposed to evaluate the importance of frames and categorize videos accordingly. In picture classification filed, CNN has shown its remarkable ability. We propose a multi-dimensional CNN algorithm, which differs from traditional conventional image processing, to represent the importance of a frame or a clip, and thus to achieve more accurate classification.

We design a hybrid-stream big data analytics model, as shown in Fig. 1, which consists of four modules: Video Pre-processing, Video Classification, Frame-Load-Reduction Processing, and Video Decision. Video Pre-processing is designed to process videos sent from the cameras. Since it is impractical to record key frames if videos are separated into many frames, we propose to divide raw video resources into two input forms: video frames and video clips. Specially, video clips are categorized by a multi-dimensional CNN. After the videos are processed by the first module, the video frames and video clips are sent to the second module called Video Classification, which calculates importance of every frame and clip. Frame-Load-Reduction Processing picks up the key frames, drops the useless frames, and decides the video's class. Finally, the Video Decision will identify an appropriate threshold to handle these videos.

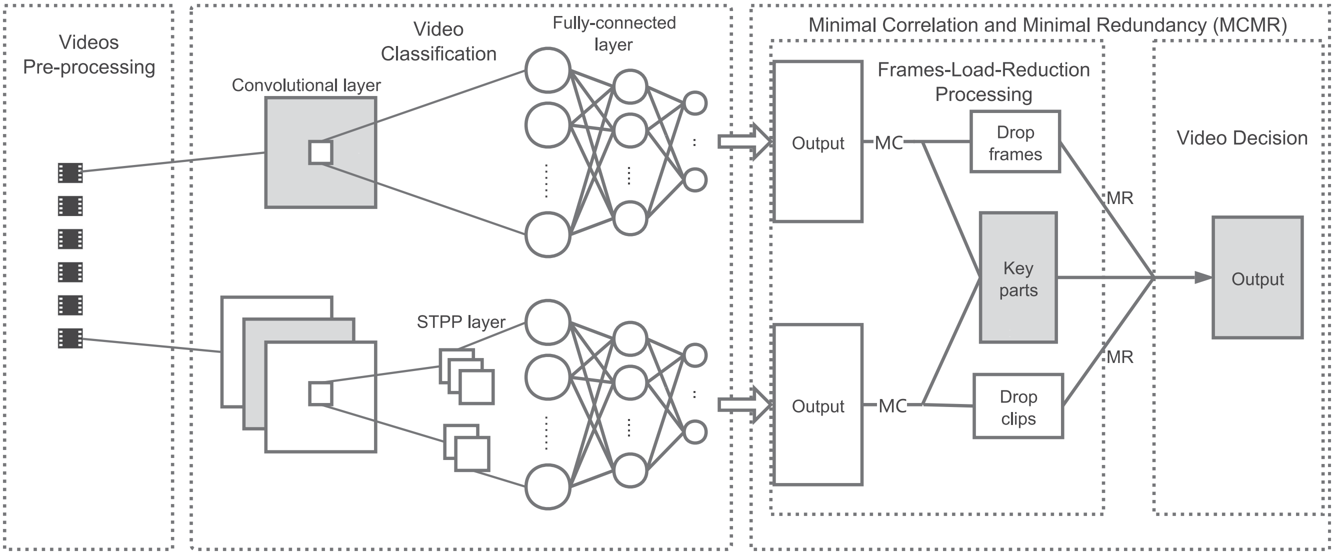


Fig. 1. Hybrid-stream big data analytics model.

3 Video Classification

3.1 Video Streams

Every CNN module consists of an input layer, an output layer, and some hidden layers. Single video image frame as an input is processed by convolution and pooling operation, and goes through fully connected layers. Finally, the network outputs a value which represents the information of this single video image frame.

This paper uses *Spatial Perceptual Information (SI)* and *Temporal perceptual Information (TI)* [22] to define information of video frames and clips. SI and TI are defined as

$$SI = max_{time}\{std_{space}[Sobel(F_n)]\}, \tag{1}$$

and

$$TI = max_{time}\{std'_{space}[M_n(i,j)]\}. \tag{2}$$

In Equation (1), $Sobel$ is a Sobel filter, F_n is the nth frame, and std_{space} represents the standard deviation over the pixels in each frame filtered by Sobel [22]. In Equation (2), TI is computed as the maximum over time (max_{time}) of the standard deviation over space (std'_{space}) of $M_n(i,j)$ over all i and j, and $M_n(i,j)$ is the difference between pixels at the same position in two subsequent frames, i.e.,

$$M_n(i,j) = F_n(i,j) - F_{n-1}(i,j), \tag{3}$$

where $F_n(i,j)$ represents the pixel at the ith row and the jth column of the nth frame in time [22].

Two types of input streams are defined, which are video frame f_i and video clip Kf_i. The K represents the length of the video clips.

In convolutional layers, the output of video frames is described as

$$f_j^{i,j} = \sum_{j=1}^{J}\sum_{i=1}^{I} f_i * \omega_{i,j}, \tag{4}$$

where $\omega_{i,j}$ denotes the weight matrices in input, I and J is the limitation in width and in height, which can be determined by convolution size [23]. Then, the output of video clips in the time dimension can be formulated as

$$f_j^{i,j,k} = \sum_{j=1}^{J}\sum_{i=1}^{I}\sum_{k=1}^{K} f_i * \omega_{i,j,k}, \tag{5}$$

where $\omega_{i,j,k}$ is the input weight matrices.

3.2 Spatio-Temporal Pyramid Pooling

The parameter K inserts time dimension into the convolution in Equation (5). However, videos may not have the same length in the time dimension, which produce outputs with different lengths. These irregular outputs are not adapted to the fixed-dimension fully connected layers. Since these fully connected layers only receive *fixed* inputs, Spatio-Temporal Pyramid Pooling (STPP) is proposed, as shown in Fig. 2. Inspired from Spatial Pyramid Pooling (SPP) [24], we add time dimension to this pooling method. In the classification module, the STPP layer is added after the last convolutional layer. The STPP layer is used to make CNN compatible with video of different lengths. It can receive different numbers of parameters as inputs and outputs fixed results to the subsequent fully connected layers. Specifically, the STPP layer pools the feature

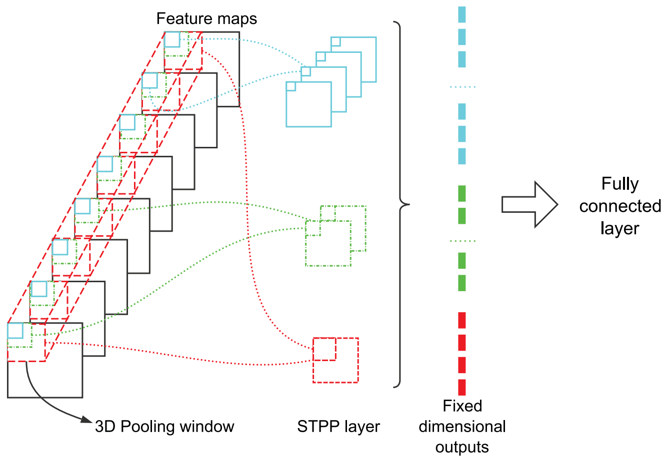


Fig. 2. Spatio-temporal pyramid pooling.

maps by using 3D pooling window and produces the output fixed-deimensional vectors, which flow into the fully connected layers.

The STPP layer can be defined as

$$[[x_1, y_1, z_1], [x_2, y_2, z_2], \ldots, [x_i, y_i, z_i]], \tag{6}$$

where $[x_i,\ y_i,\ z_i]$ represents the width, the length and the time dimensional metric of the ith part of the STPP layer, respectively. The output dimension of the STPP layer D is calculated as

$$D = \sum_{i=0}^{I} x_i y_i z_i. \tag{7}$$

Assuming there are π feature maps of size $m \times n$ as an input, the 3-dimensional pooling window can be formulated as

$$(P_{x_i}, P_{y_i}, P_{z_i}) = \left(\left\lfloor \frac{m}{x_i} \right\rfloor, \left\lfloor \frac{n}{y_i} \right\rfloor, \left\lfloor \frac{\pi}{z_i} \right\rfloor \right), \tag{8}$$

where $P_{x_i}, P_{y_i}, P_{z_i}$ represent the width, the length and the time dimensional metric of the ith pooling window in the STPP layer, respectively.

3.3 Training the Multi-Dimensional CNN with the Spatio-Temporal Pyramid Pooling Layer

Theoretically, our classifier can be trained by using the standard back-propagation. The cost function is defined as

$$J(w, b) = \frac{1}{2} \| h_{w,b}(x) - \alpha \|^2, \tag{9}$$

where w denotes the collections of all weights in the network, b denotes all the biases, and α is the desired output from the network.

In proposed neural network, the output's dimension needs to be determined by the actual condition. We use $h_{w,b}(x)$, which is composite by nodes of the network, to represent the network output of the input x. The notation $\| \cdot \|$ just denotes the norm for a vector.

From the quadratic cost function, it can be seen that $J_{w,b}$ is non-negative. In other words, $J_{w,b} \geq 0$ and it becomes small, i.e., $J_{w,b} \approx 0$, when $y(x)$ is approximately equal to the training output, a, for all the inputs x. Thus, the suitable weights and biases will be found to make $J_{w,b} \approx 0$. On the contrary, it is not ideal when the cost function $J_{w,b}$ is large, which means α is not equal to the output $h_{w,b}(x)$ for numerous inputs x. To find the set of weights and biases to minimize the cost $J_{w,b}$, we proposed to use the stochastic gradient descent (SGD) method [25], i.e., J varies as

$$\Delta J \approx \frac{\partial J}{\partial w} \Delta w + \frac{\partial J}{\partial b} \Delta b. \tag{10}$$

Note that

$$\Delta J = \nabla J \bullet (\triangle w, \triangle b), \tag{11}$$

where ∇J is the gradient vector,

$$\nabla J = \left(\frac{\partial J}{\partial w}, \frac{\partial J}{\partial b} \right)^T. \tag{12}$$

Denote η as the learning rate, which is a small and positive parameter. Thus,

$$(\triangle w, \triangle b) = -\eta \nabla J = -\eta \left(\frac{\partial J}{\partial w}, \frac{\partial J}{\partial b} \right)^T, \tag{13}$$

and

$$\Delta J = -\eta \| \nabla J \|^2 = -\eta \| \left(\frac{\partial J}{\partial w}, \frac{\partial J}{\partial b} \right)^T \|^2 \leq 0. \tag{14}$$

Therefore, J always decreases with respect to w and b. By separating W and b, the following two equations can be derived as

$$w_{k+1} = w_k - \eta \frac{\partial J}{\partial w_k}, \tag{15}$$

and

$$b_{k+1} = b_k - \eta \frac{\partial J}{\partial b_k}. \tag{16}$$

To simplify the calculation, we set m as

$$\frac{\sum_{j=1}^{m} \nabla J_{xi}}{m} \approx \frac{\sum_x \nabla J_x}{n} = \nabla J, \tag{17}$$

$$w_{k+1} = w_k - \frac{\eta}{m} \frac{\partial J}{\partial w_k}, \tag{18}$$

and

$$b_{k+1} = b_k - \frac{\eta}{m} \frac{\partial J}{\partial b_k}. \tag{19}$$

4 Frame-Load-Reduction and Video Decision

In this section, the Frame-Load-Reduction Processing and Video Decision are introduced. These two modules are the key to deciding the storage class. In the Frame-Load-Reduction, some less important frames will be dropped. However, if some significant frames are dropped, it may lead to the decrease of video quality. Therefore, the Video Decision module is designed to reduce the probability of this situation.

4.1 Key Frames

The information values of video frames and video clips are the two inputs to the Frame-Load-Reduction module. We use the outputs of the two video abstract mechanisms in Section 3 instead of original video as input, because the information values of video frames and video clips can abstract the intrinsic properties of the original video, and can be readily used to analyze the video. We propose using a set of frames, K, to represent the key frames as follows:

$$K = F_{k-frames}(Video) = \{ f_{r_i} | i = 1, 2, \ldots, I \}, \tag{20}$$

where I is the count of key frames and f_{r_i} is a key frame set in the inputs. $F_{k-frames}$ is the function that operates on the original video to generate these key frames. Similarly, F_{skim} is used to represent the set of "skim" videos, a set of video clips extracted from the original video to represent a very short synopsis of the original video. The set of "skim" videos, S, is defined as

$$S = F_{skim}(Video) = f_{r_1} \bigodot f_{r_2} \bigodot \cdots \bigodot f_{r_i}, \tag{21}$$

where $\bigodot$ is the operation of joining video clips.

4.2 Key Frame Ratio

In the important feature extraction, we set a ratio θ to reduce the frames.

Finding the threshold can be formulated as an optimization problem of finding a suitable set $R = (q_1, q_2, \ldots, q_j)$, which can represent the original video with the least number of frames or clips, i.e.,

$$\begin{cases} \{q_1, q_2, \ldots, q_j\} = \arg\min_{q_j} \{D(R, F) | 1 \leq q_j \leq n\} \\ j = \theta \cdot n, \end{cases} \tag{22}$$

where n is the number of frames or clips in the original video sequence, j is the count of key frames or clips, D is a dissimilarity measurement, and F is the output of the classification module.

4.3 Minimal Correlation Among Key Frames or Video Clips

Then, we start to find the suitable set to represent the video with the least number of frames or clips. The minimal correlation method selects frames or clips that are dissimilar to each other. In other words, this method can represent the video with the least elements. After introducing the concept of minimal correlation, Equation (22) can be rewritten as

$$\{q_1, q_2, \ldots, q_j\} = \arg\min_{q_j} \{Corr(f_{r_1}, f_{r_2}, \ldots, f_{r_i})\}, \tag{23}$$

where $Corr$ is the correlation function.

In order to complete the $Corr$ operation, pairs of frames or clips are used as

$$Corr(f_{r_1}, f_{r_2}, \ldots, f_{r_i}) = \left\{ \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} Corr(f_{r_i}, f_{r_j})^2 \right\}^{1/2}, \tag{24}$$

where $Corr(f_{r_i}, f_{r_j})$ is the correlation coefficient of any two frames or clips (f_{r_i}, f_{r_j}), and K represents the length of video clips, defined by Equation (5). In this paper, sequential elements are taken into consideration, and the Equation (24) can be rewritten as

$$\{q_1, q_2, \ldots, q_j\} = \arg\min_{q_j} \left\{ \sum_{i=1}^{\sigma} Corr(f_{r_i}, f_{r_{i+1}}) \right\}, \tag{25}$$

where σ is the count of key frames.

However, the aim of extracting key frames or clips is to maximize the difference between each pair of frames or clips, rather than simply reduce the total number of key frames.

The pseudo-codes of scene change and key frame extraction in the load-reduction module are presented in Algorithm 1. In this algorithm, A denotes the scene change threshold, and λ denotes the dynamic load-reduction threshold. As the scene changes, a different scene may require a different threshold in signifying a scene change. Because Algorithm 1 goes through every frames, the computation and space complexities of this algorithm are $O(n)$ and $O(1)$, respectively. Final performance can be improved by recognizing scenes correctly. The output set S is our processed video.

Algorithm 1. Scene Change and Key frame Extraction

```
Input: f_{r_i}, A, λ
Output: S
begin
    while (i < n) do
        if (Corr(f_{r_i}) < A) then
            enter into scene + 1
        else
            still in scene
        end
        while scene has not change do
            if (f_{r_i} > λ) then
                drop the frame
            else
                store the frame in set S
            end
        end
    end
end
```

4.4 Minimal Redundancy

In this part, another ratio μ is set to decide the frames to be dropped. In discarding frames and clips, we need to ensure not to drop important frames. Therefore, the ratio μ should be appropriately set to improve the accuracy of the dropping algorithm.

A set $T = \{p_1, p_2, \ldots, p_v\}$ is proposed to represent the dropped video with the least number of frames or clips. The optimization problem can be formulated as

$$\begin{cases} \{p_1, p_2, \ldots, p_v\} = \arg\min_{p_v} \{R(T, F_r) | 1 \leq p_v \leq n_r\} \\ v = \mu \cdot n_r, \end{cases} \tag{26}$$

where n_r is the number of dropped frames or clips in the original video sequence, v is the count of similar frames or clips, F_r is the part of the output of the Frame-Load-Reduce module to be dropped, and R is a similarity measure which is formulated as

$$R = \sum^{P} |\triangle pixelvalue|, \tag{27}$$

where P represents all the pixel of a frame or a clip, $\triangle pixelvalue$ is the difference between two corresponding pixel value.

The minimal redundancy is used to find the set. Specifically, it is to select the dropped frames or clips that are similar to each other. So, Equation (26) can be rewritten as

$$\{p_1, p_2, \ldots, p_v\} = \arg\min_{p_v} \{Sim(f_{r_1}, f_{r_2}, \ldots, f_{r_i})\}, \tag{28}$$

$$Sim(f_{r_1}, f_{r_2}, \ldots, f_{r_i}) = \left\{ \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} Sim(f_{r_i}, f_{r_j})^2 \right\}^{\frac{1}{2}}, \tag{29}$$

$$\{p_1, p_2, \ldots, p_v\} = \arg\min_{p_v} \left\{ \sum_{i=1}^{\sigma} Sim(f_{r_i}, f_{r_{i+1}}) \right\}, \tag{30}$$

where $Sim(f_{r_1}, f_{r_2}, \ldots, f_{r_i})$ is the similarity measure of frames or clips $\{f_{r_1}, f_{r_2}, \ldots, f_{r_i}\}$, and K represents the length of the video clips, defined in Equation (5).

4.5 Videos Decision

Once θ and μ are determined, the limit to decide videos can be calculated. Our goal is to divide these videos into two classes, retaining the original videos or the processed videos. The limit is formulated as

$$class = \begin{cases} \text{original videos}, & (\theta + \mu < threshold) \\ \text{processed videos}, & (\theta + \mu \geq threshold), \end{cases} \tag{31}$$

where $threshold \in [0, 2]$. Since lower video quality loss is one of our system design goals, $threshold$ should be carefully chosen. First, we define video quality loss function Q using *SI* and *TI* as

$$\begin{cases} Q(video, threshold) = \Delta SI + \Delta TI \\ \Delta SI = SI_{video} - SI_{video'} \\ \Delta TI = TI_{video} - TI_{video'}, \end{cases} \tag{32}$$

where SI indicates the spatial detail quantity in a frame, and TI indicates the number of temporal changes of a video sequence [22]. Function Q calculates $video'$ from original $video$ using the whole system. Then, choosing $threshold$ can be formulated as an optimization problem

$$threshold = \arg \min_{threshold} \{Q(video, threshold)\}. \tag{33}$$

In this paper, a Genetic Algorithm, as shown in Algorithm 2, is proposed to determine the $threshold$.

Algorithm 2. Decrease the Loss of Video Quality

Input: P_c, P_m, M, Q, S
Output: $threshold$
begin
  Random initialize 1th generation $parent = \{i_1, i_2, \ldots, i_M\}$
  while $True$ **do**
    Calculate $Q_{parent} = \{Q(i_1), Q(i_2), \ldots, Q(i_M)\}$
    $child = \emptyset$
    for $Size_i < M$ **do**
      $a, b = S(Q_{parent})$
      if $P_c > random(0, 1)$ **then**
        $a, b = Crossover(a, b)$
      end
      if $P_m > random(0, 1)$ **then**
        $a, b = Mutation(a, b)$
      end
      Append a, b to set $child$
    end
    if $Max(Q_{child}) < Max(Q_{parent})$ **then**
      $threshold = Max(Q_{parent})$
      Output $threshold$
      Break
    end
    $parent = child$
  end
end

In this algorithm, P_m is the mutation probability, P_c is the crossover probability, M is the number of individuals in every generation, Q is the video quality loss function in Equation (32), and S represents select operation. The algorithm starts with creating the 1th generation randomly. The GA algorithm iterates and gets the optimal $threshold$ gradually in every generation, GA generates $child$ randomly by crossover or mutation, then it chooses the best one in $child$ and $parent$ to perform the next iteration. When GA finishes, the optimal $threshold$ will be given. This $threshold$ is used to control the videos decision, leading to the increment of classification accuracy. Moreover, the non-redundant video will not be dealt with. The computation and space complexities are $O(n^2)$ and $O(MP_m)$, respectively. Because the threshold θ is determined by GA, it makes fewest video quality loss and responds well to the influx of data.

5 Performance Evaluations

In this section, the performance of the proposed hybrid-stream model is verified. First, the data set we use in this model and different architectures in pooling operation are listed, respectively. After that, we conduct a series of experiment to analyze the performance of classification module, load reduction module, and video decision.

5.1 Experimental Setup

In this section, a simulation model is developed based on a data center who has the same architecture with [27] to estimate our proposed model. The number of groups in these experiments varies from 50 to 250 and the number of joining or leaving nodes selected from the groups varies from 100 to 1000. As shown in Table 1, our simulation is mainly carried out on eight data sets. Aimed at reducing the multimedia data redundancy in data centers, we choose these data sets that consist of enough videos.

To study the influence of different architectures of STPP layers on our frame classification module, we set four different architectures of STPP layers, which are clarified in Table 2. For comparison, the performance of following algorithms, which are also used in video load reduction [14], are also shown as benchmarks.

- *k-Nearest Neighbor (KNN):* The KNN is a classifier that classifies the samples by the votes of k-nearest neighbor [14].
- *Support Vector Machine (SVM):* It uses kernel and hyperplane technical to classify the non-linear samples [14].
- *Basic CNN (LeNet* [14]): The CNN is a classifier which is good at processing image, it uses convolutional operation to abstract the information in image.

These benchmarks are used to analyze the performance of the proposed STPP, Algorithms 1 and 2.

The training is performed by Keras and TensorFlow, the loss function is Mean-Squared-Error, and the optimizer is Adam. All the parameters are default. The metrics to evaluate the performance of classifier are accuracy and Receiver Operating Characteristic (ROC) which is a kind of curve illustrating the performance of a classifier system. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. If

TABLE 1
Detailed Data Sets in Simulation

Data set	Content	Size	Format	Resolution
Alpha[a]	Dashboard camera videos	100 GB	MP4	480×320
Beta	Open class videos	60 GB	AVI	750×540
Gamma	College surveillance videos	200 GB	MP4	720×480
Delta[b]	Videos of crowd	20 GB	MP4	640×480
Epsilon[c]	Soccer videos	100 GB	RMVB	1280×720
Zeta[d]	Aerial photography	10 GB	MP4	1920×1080
Eta[e]	Video call	100 GB	AVI	1280×720
UCF-101 Expand[f]	Extension of UCF-101	18 GB	AVI	140×100, 80×60

a. Generated by dashboard cameras in authors's cars.
b. Beta, Gamma and Delta are provided by Nanjing University of Posts and Telecommunications.
c. Chinese Football Association Super League.
d. Generated by author's unmanned aerial vehicle (DJI phantom-4).
e. Authors' group meeting.
f. UCF-101 Expand is an extension of UCF-101 [26]. Some transformations, such as overexposure and pattern distortion, are performed in UCF-101 to create new videos that are close to real data centers' working environment. Then, we get UCF-101 Expend by merging these new videos to UCF-101. The size of UCF-101 Expand is about 18 GB.

there is a perfect classifier, its ROC curve will all locate in (0, 1), which is called perfect classification.

5.2 Experimental Results

5.2.1 The Influence of Frame Classification

Since the Frame-Load-Reduction Processing and Video Decision are based on the frame classification, we hold that the performance of classifier can strongly influence the performance of data redundancy reduction. In order to investigate the relation between classification and data redundancy reduction, we conduct a series of experiments. The results of experiments are shown in Fig. 3, revealing that the redundancy reduction rate has obviously positive correlation with the accuracy of classification, i.e., the improvement of the accuracy of classification helps reduce the data redundancy.

5.2.2 The Performance of Multi-Dimensional CNN Classification

As shown in Fig. 4, our classification method (CNN with 3D pooling, which is indicated by 3D CNN) has basically higher accuracy than other models. The STPP layer allows this module to accept various resolution videos. In other words, the hybrid-stream model does not require cropping or scaling the input video to the fixed resolution. Cropping and scaling usually mean the loss of video information. We further explore the influence of different architectures of STPP layers. The classification module in four pooling architectures performs in UCF-101 Expand. Comparison is also made to prove that our method has a good performance. Fig. 5 shows the performance of four STPP architectures with basic and multi-dimensional convolution. The multi-dimensional convolution improves the classification accuracy obviously. The ROC curves of our multi-dimensional classification module in different pooling layer architectures are presented in Fig. 6. Because of more pyramid parts in pooling layer and reasonable dimensional metric, architecture **C** is closer to the

TABLE 2
Different STPP Architectures in Simulation

	Architectures				
Type	$x_1 \times y_1 \times z_1$[a]	$x_2 \times y_2 \times z_2$	$x_3 \times y_3 \times z_3$	$x_4 \times y_4 \times z_4$	Outputs
A	$5 \times 3 \times 2$	$7 \times 5 \times 3$	$14 \times 10 \times 5$	$1 \times 1 \times 1$	835
B	$1 \times 1 \times 2$	$2 \times 2 \times 2$	$3 \times 2 \times 3$	$1 \times 1 \times 1$	102
C	$2 \times 2 \times 3$	$3 \times 3 \times 4$	$3 \times 4 \times 7$	$1 \times 1 \times 1$	132
D	$1 \times 1 \times 3$	$2 \times 1 \times 3$	$4 \times 4 \times 2$	$10 \times 6 \times 8$	401

a. x_i, y_i, z_i represent the width, the length and the time dimensional metric of the ith part in STPP layer, where $i \in \{1, 2, 3, 4\}$ is the index of STPP layer parts. Outputs are fixed and calculated by Equation (7).

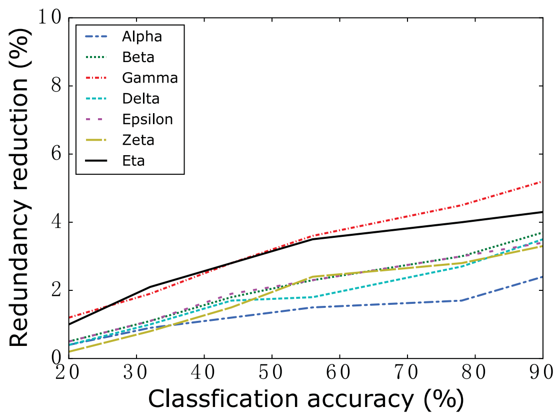


Fig. 3. The redundancy reduction rate of different data sets.

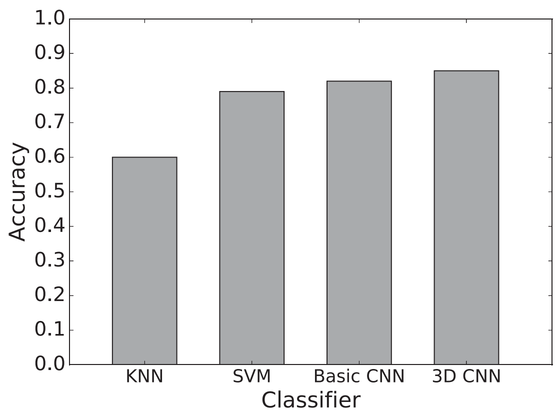


Fig. 4. Comparison among four classifiers.

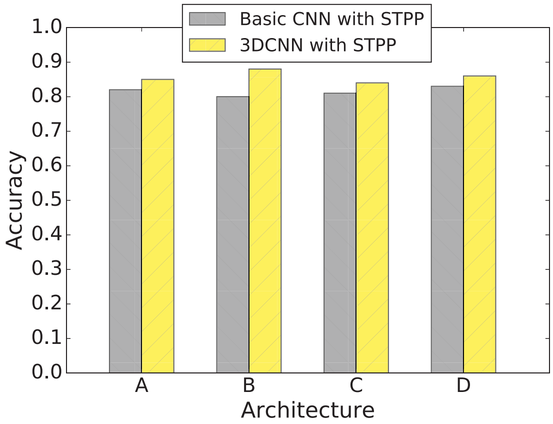


Fig. 5. Comparison between two CNNs in different STPP architectures.

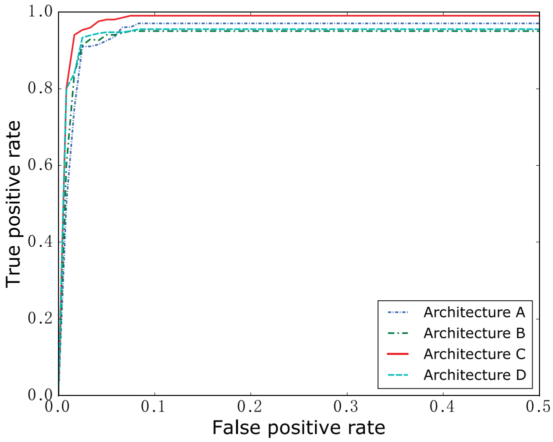


Fig. 6. Comparison among different STPP architectures.

TABLE 3
Thresholds in Eight Data Sets with Four STPP Architectures

	Alpha	Beta	Gamma	Delta	Epsilon	Zeta	Eta	UCF-101 Expand
A	1.32	1.17	1.22	1.13	0.87	1.42	1.44	0.96
B	1.26	1.05	1.35	1.05	0.92	1.31	1.52	1.05
C	1.13	1.12	1.27	1.18	0.60	1.29	1.38	0.92
D	1.35	1.00	1.01	1.11	0.88	1.34	1.35	1.10

upper left corner, which represents better classification performance.

5.2.3 The Impacts of Frame-Load-Reduction and Video Decision

The performance of Frame-Load-Reduction and Video Decision modules are illustrated. We perform GA to determine the *threshold*, and the *thresholds* in eight data sets with four STPP architectures are shown in Table 3.

To clarify the capability of MCMR algorithm and the average performance of our model in different data sets, size reduction before (SRB) and after (SRA) performing MCMR are shown in Table 4. Different data sets have different reduction rates under our model and the average redundancy-reduction percentage is about 3.7 percent. Owing to MCMR algorithm that sacrifices a little size of reduction, videos generally have considerable objective quality assessment promotion, which is evaluated by TI and SI.

TABLE 4
Redundancy-Reduction in Different Types of Video

Data set	SRB	SRA	Objective Quality Assessment promotion
Alpha	3.3%	2.4%	13%
Beta	4.1%	3.7%	7%
Gamma	5.8%	5.2%	9%
Delta	4.3%	3.5%	15%
Epsilon	5.8%	3.4%	16%
Zeta	4.4%	3.3%	10%
Eta	6.4%	4.3%	14%
UCF-101 Expand	4.5%	4.0%	5%
Average	4.8%	3.7%	11%

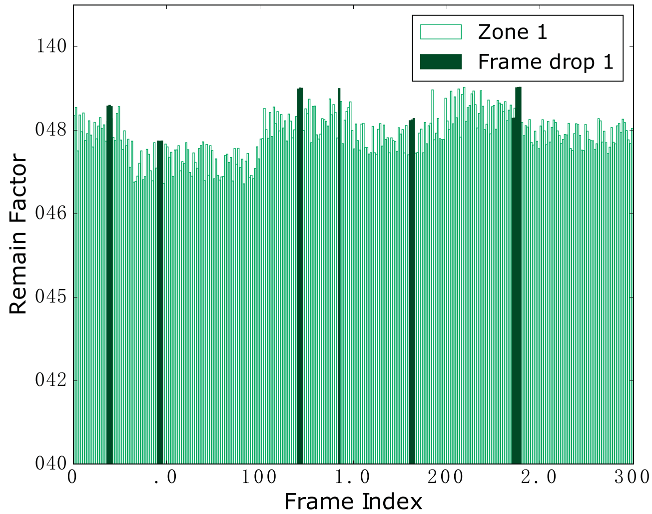


Fig. 7. One scene in a 300-frames video and its drop frames.

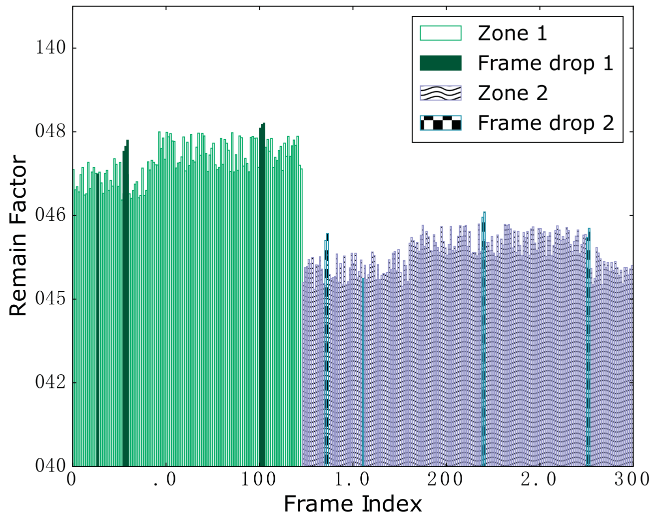


Fig. 8. Two scenes in a 300-frames video and its drop frames.

Figs. 7 and 8 show the performance of load reduction module. The horizontal axis in two figures are both corresponding to a 10-frames video clip extracted from two different scenes. Fig. 7 shows that six frame clusters are dropped in one scene. There is an obvious scene change in Fig. 8, where different colors are used to distinguish between two scenes. From Figs. 7 and 8, the dropping of frame clips can be visually observed.

6 Related Works

6.1 Storage Load for Multimedia Big Data

Actually, studies on storage system load focus on load models and load balancing strategies. Typically, strategies determine the load based on various properties of the node. Lu et al. [28] proposed a hybrid control strategy for load balancing, which is triggered by the load and accessing latency. Hong et al. [29] studied the problem of visualize storage node. Wei et al. [30] improved dynamic storage balancing using resource popularity prediction, which makes less replication. Liu et al. [12] proposed a strategy for load balancing in virtual storage, it has a writing balancing module which calculates the weight of each storage node to get the best one. This module gathers six parameters, which are CPU Percent Utilization, Memory Percent Utilization, Net Flow, Disk I/O FQCY, Response Time, Process Sum and evaluating server's load. LBVS uses Weight Parameter π_i to adjust the proportion of parameters, and the algorithm chooses the best node according to the load. Tan et al. [31] proposed Duplex Loading Balancing Strategy on Object Storage System (DLBS). DLBS actually has two modules including Active Balancing Strategy (ABS) and Passive Balancing Strategy (PBS). ABS uses greedy algorithm to balance the load performance during the whole process of three critical steps, and PBS controls the replantation of host replicas and the drop operation of redundant ones. Prabavathy et al. [17] used a dynamic weight decided by network and disk usage rate, and also a more detailed load model including eight parameters. This proposed model strives to balance every node's disk usage rate, according to the data migration algorithm which acts on a pair of nodes and has various load status. Zhou et al. [18] proposed Optimize Block-Level Cloud Storage System with Load-Balance Strategy (BCSLS), where it transforms the goal of load balance into making the range of nodes' load smaller than the threshold.

6.2 Redundancy Avoidance in Data Center

Data center has emerged as a crucial infrastructure that holds ever-growing severs. Tens or hundreds of thousands of service and applications are hosted in modern data centers. Recently works [32], [33], [34] summarized the challenges and requirements for data centers' design and operations in several aspects: large scale, wide variety of applications, high energy consumption, and strict service requirement. Faced with the aforementioned technical challenges, data center has drawn significant attention on achieving various objectives, such as low power consumption, high utilization, quality guarantee, and high robustness.

To reduce the storage load and increase the service quality of data centers, the usual practice was to reduce the redundancy of the content stored in data center. Peter et al. [35] gave a survey of indexing techniques for data deduplication. The popular study in this field was based on similar searching. Hu et al. [36] studied the similar search problem with Maximum Common Connected Subgraph (MCCS) constraints, and proposed a framework based on edge matching situation to solve this problem. Mao et al. [37] investigated the probability of optimizing the read-performance by hardware. They proposed a Solid-State Drive (SSD)-assisted read scheme which could accelerate the speed of similar searching. The fundamental concept of this read scheme is the high random access speed of SSD. Li et al. [38] considered the differential privileges of users in similar searching as well as the data itself. They further proposed a distributed reliable deduplication system which focuses on the security in distributed data centers [39]. This system was achieved by storing the hash tag value in every storage data center, and was able to guarantee the consistency of tags. Kolb et al. [40] proposed an approach which does not rely on re-clustering, and is capable of parallel MapReduce processing. Costa et al. [41] proposed an incremental clustering algorithm, which was the first to regard increment as a major requirement in data deduplication. Different from the studies above, Kaaniche et al. [42] proposed a client side deduplication scheme. It carefully controlled the privileges of unauthorized users. Since the process of encrypting the data is performed in client side, the computational resource occupied by encryption in data centers is reduced.

Previous studies on storage load reduction in data centers mainly focus on load balancing strategies and flow control. They are not appropriate for processing multimedia data in storage system because algorithms redistribute data, but not reduce data redundancy. In other words, the load of the whole system is not reduced in real sense. To our best knowledge, we are the first to use multi-dimensional video information for reducing the storage load in data centers.

7 Conclusions

In this paper, a hybrid-stream big data analytics model has been proposed to enhance the classification precision and relieve the data centers' network and storage overload. The model can improve the speed to deal with the videos and recognizing, deciding the important frames and whether to drop the unimportant ones in every video. Compared to conventional methods like deep learning to address image analysis problems, this paper has improved the method to deal with video analysis. Besides, this network and storage overload problem of video is considered as an optimization problem, which can show a practical algorithm over a large-scale of real-time data from numerous nodes. The conducted simulations represent that our model performs well in most of the data sets. Moreover, the hybrid-stream big data analytics model and the improved video with recognized algorithm can lead to a fairly good video stream and save storage space in the Internet of Things. Our algorithm also provides a way to relieve the network and storage load. The model can reduce network and storage overload, and it will not destroy the truly important videos as well.

Acknowledgments

This work is supported by NSFC (61572262, 61772286), the China Postdoctoral Science Foundation (2017M610252), the China Postdoctoral Science Special Foundation (2017T 100297), the Open Research Fund of the Jiangsu Engineering Research Center of Communication and Network Technology, NJUPT; and the National Engineering Research Center of Communications and Networking (Nanjing University of Posts and Telecommunications)(TXKY17014).

REFERENCES

[1] J. Wu, I. Bisio, C. Gniady, E. Hossain, M. Valla, and H. Li, “Context-aware networking and communications: Part 1 [guest editorial],” *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 14–15, Jun. 2014.

[2] 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.

[3] C. Ge, Z. Sun, N. Wang, K. Xu, and J. Wu, “Energy management in cross-domain content delivery networks: A theoretical perspective,” *IEEE Trans. Netw. Serv. Manage.*, vol. 11, no. 3, pp. 264–277, Sep. 2014.

[4] K. Wang, Y. Shao, L. Shu, C. Zhu, and Y. Zhang, “Mobile big data fault-tolerant processing for ehealth networks,” *IEEE Netw.*, vol. 30, no. 1, pp. 36–42, Jan. 2016.

[5] H. Jiang, K. Wang, Y. Wang, M. Gao, and Y. Zhang, “Energy big data: A survey,” *IEEE Access*, vol. 4, pp. 3844–3861, 2016.

[6] L. Gu, D. Zeng, S. Guo, Y. Xiang, and J. Hu, “A general communication cost optimization framework for big data stream processing in geo-distributed data centers,” *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 19–29, Jan. 2016.

[7] Z. B. Liu, “The realization of virtual storage networking based on asymmetric architecture,” *Comput. Sci.*, vol. 31, no. 6, pp. 52–55, 2004.

[8] C. F. Lai, Y. X. Lai, M. S. Wang, and J. W. Niu, “An adaptive energy-efficient stream decoding system for cloud multimedia network on multicore architectures,” *IEEE Syst. J.*, vol. 8, no. 1, pp. 194–201, Mar. 2014.

[9] J. Wu, S. Guo, J. Li, and D. Zeng, “Big data meet green challenges: Big data toward green applications,” *IEEE Syst. J.*, vol. 10, no. 3, pp. 888–900, Sep. 2016.

[10] K. Wang, Y. Wang, X. Hu, Y. Sun, D. J. Deng, A. Vinel, and Y. Zhang, “Wireless big data computing in smart grid,” *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 58–64, Apr. 2017.

[11] K. Wang, H. Li, Y. Feng, and G. Tian, “Big data analytics for system stability evaluation strategy in the energy internet,” *IEEE Trans. Ind. Inform.*, vol. 13, no. 4, pp. 1969–1978, Aug. 2017.

[12] H. Liu, S. Liu, X. Meng, C. Yang, and Y. Zhang, “LBVS: A load balancing strategy for virtual storage,” in *Proc. Int. Conf. Serv. Sci.*, May 2010, pp. 257–262.

[13] S. Lee, K. Sriram, K. Kim, Y. H. Kim, and N. Golmie, “Vertical handoff decision algorithms for providing optimized performance in heterogeneous wireless networks,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 865–881, Feb. 2009.

[14] K. Wang, J. Mi, C. Xu, Q. Zhu, L. Shu, and D.-J. Deng, “Real-time load reduction in multimedia big data for mobile internet,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 5s, pp. 76:1–76:20, Oct. 2016. [Online]. Available: http://doi.acm.org/10.1145/2990473

[15] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 568–576. [Online]. Available: http://dl.acm.org/citation.cfm?id=2968826.2968890

[16] H. Ye, Z. Wu, R.-W. Zhao, X. Wang, Y.-G. Jiang, and X. Xue, “Evaluating two-stream CNN for video classification,” in *Proc. 5th ACM Int. Conf. Multimedia Retrieval*, 2015, pp. 435–442. [Online]. Available: http://doi.acm.org/10.1145/2671188.2749406

[17] B. Prabavathy, K. Priya, and C. Babu, “A load balancing algorithm for private cloud storage,” in *Proc. 4th Int. Conf. Comput. Commun. Netw. Technol.*, Jul. 2013, pp. 1–6.

[18] L. Zhou, Y. C. Wang, J. L. Zhang, J. Wan, and Y. J. Ren, “Optimize block-level cloud storage system with load-balance strategy,” in *Proc. IEEE 26th Int. Parallel Distrib. Process. Symp. Workshops PhD Forum*, May 2012, pp. 2162–2167.

[19] M. Baktashmotlagh, M. Harandi, B. C. Lovell, and M. Salzmann, “Discriminative non-linear stationary subspace analysis for video classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2353–2366, Dec. 2014.

[20] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, “Green resource allocation based on deep reinforcement learning in content-centric IoT,” *IEEE Trans. Emerging Topics Comput.*, to be published, doi: 10.1109/TETC.2018.2805718.

[21] 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, Dec. 2016.

[22] Methodology for the subjective assessment of video quality in multimedia applications. International Telecommunication Union (ITU). Jan. 2007. [Online]. Available: http://www.itu.int/rec/R-REC-BT.1788-0-200701-I/en

[23] K. Hamedani, L. Liu, R. Atat, J. Wu, and Y. Yi, “Reservoir computing meets smart grids: Attack detection using delayed feedback networks,” *IEEE Trans. Ind. Inform.*, vol. 14, no. 2, pp. 734–743, Feb. 2018.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[25] L. Bottou and O. Bousquet, “The tradeoffs of large scale learning,” in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 161–168.

[26] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild,” Univ. Central Florida, FL, Tech. Rep. CRCV-TR-12-01, Nov. 2012.

[27] C. C. Chuang, Y. J. Yu, A. C. Pang, H. W. Tseng, and H. P. Lin, “Efficient multicast delivery for data redundancy minimization over wireless data centers,” *IEEE Trans. Emerging Topics Comput.*, vol. 4, no. 2, pp. 225–241, Apr. 2016.

[28] Y. Lu, J. Zhang, S. Wu, and S. Zhang, “A hybrid dynamic load balancing approach for cloud storage,” in *Proc. Int. Conf. Ind. Control Electron. Eng.*, Aug. 2012, pp. 1332–1335.

[29] H. Tao, W. Yating, C. Bingyao, Y. Ke, and Y. Fei, “A dynamic data allocation method with improved load-balancing for cloud storage system,” in *Proc. IET Int. Conf. Smart Sustainable City*, Aug. 2013, pp. 220–225.

[30] X. Wei, J. Yang, and H. Xi, “A novel content updating policy based on dynamic storage balancing for clustered streaming media system,” in *Proc. IEEE Int. Conf. Control Autom.*, Dec. 2009, pp. 2031–2036.

[31] T. Zhipeng, F. Dan, T. Xudong, and H. Fei, “DLBS: Duplex loading balancing strategy on object storage system,” in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl.*, Aug. 2009, pp. 45–52.

[32] W. Xia, P. Zhao, Y. Wen, and H. Xie, “A survey on data center networking (DCN): Infrastructure and operations,” *IEEE Commun. Surv. Tut.*, vol. 19, no. 1, pp. 640–656, Jan.–Mar. 2017.

[33] C. Xu, K. Wang, P. Li, R. Xia, S. Guo, and M. Guo, “Renewable energy-aware big data analytics in geo-distributed data centers with reinforcement learning,” *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 205–215, Jan.-Mar. 2020.

[34] 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. IEEE/ACM 25th Int. Symp. Quality Service*, Jun. 2017, pp. 1–6.

[35] P. Christen, “A survey of indexing techniques for scalable record linkage and deduplication,” *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 9, pp. 1537–1555, Sep. 2012.

[36] H. Hu, G. Li, and J. Feng, “Fast similar subgraph search with maximum common connected subgraph constraints,” in *Proc. IEEE Int. Congress Big Data*, Jun. 2013, pp. 181–188.

[37] B. Mao, H. Jiang, S. Wu, Y. Fu, and L. Tian, “Read-performance optimization for deduplication-based storage systems in the cloud,” *Trans. Storage*, vol. 10, no. 2, pp. 6:1–6:22, Mar. 2014. [Online]. Available: http://doi.acm.org/10.1145/2512348

[38] J. Li, X. Chen, X. Huang, S. Tang, Y. Xiang, M. M. Hassan, and A. Alelaiwi, “Secure distributed deduplication systems with improved reliability,” *IEEE Trans. Comput.*, vol. 64, no. 12, pp. 3569–3579, Dec. 2015.

[39] J. Li, Y. K. Li, X. Chen, P. P. C. Lee, and W. Lou, “A hybrid cloud approach for secure authorized deduplication,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1206–1216, May 2015.

[40] L. Kolb, A. Thor, and E. Rahm, “Don’t match twice: Redundancy-free similarity computation with MapReduce,” in *Proc. 2nd Workshop Data Analytics Cloud*, 2013, pp. 1–5. [Online]. Available: http://doi.acm.org/10.1145/2486767.2486768

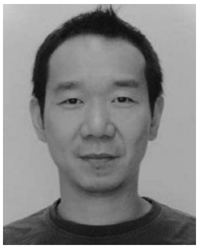
[41] G. Costa, G. Manco, and R. Ortale, “An incremental clustering scheme for data de-duplication,” *Data Mining Knowl. Discovery*, vol. 20, no. 1, pp. 152–187, Jan. 2010. [Online]. Available: http://dx.doi.org/10.1007/s10618-009-0155-0

[42] N. Kaaniche and M. Laurent, “A secure client side deduplication scheme in cloud storage environments,” in *Proc. 6th Int. Conf. New Technol. Mobility Security*, Mar. 2014, pp. 1–7.

Chenhan Xu is working toward the undergraduate degree in the School of Internet of Things, Nanjing University of Posts and Telecommunications, China. His current research interests include big data, cloud computing, blockchain, and machine learning.

Kun Wang (M'13-SM'17) received the BEng and PhD degrees in computer science from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004 and 2009, respectively. From 2013 to 2015, he was a postdoc fellow with Electrical Engineering Department, University of California, Los Angeles (UCLA), CA, USA. In 2016, he was a research fellow with the School of Computer Science and Engineering, The University of Aizu, Aizu-Wakamatsu City, Fukushima, Japan. He is currently a research fellow with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China, and also a full professor with the School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China. He has published more than 100 papers in referred international conferences and journals. He has received Best Paper Award at IEEE GLOBECOM16. He serves as associate editor of the *IEEE Access*, editor of the *Journal of Network and Computer Applications*, the *Journal of Communications and Information Networks*, the *EAI Transactions on Industrial Networks and Intelligent Systems* and guest editors of the *IEEE Access*, the *Future Generation Computer Systems*, the *Peer-to-Peer Networking and Applications*, and the *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 the IEEE and member of the ACM.

Yanfei Sun received the PhD degree in communication and information system from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2006. He has been a professor with the College of Telecommunication and Information Engineering, Nanjing University of Posts and Telecommunications, since 2006. His main research interests include the areas of future network, industrial Internet, big data management, and analysis of intelligent optimization and control.



Song Guo (M'02-SM'11) received the PhD degree in computer science from the University of Ottawa and was a professor with the University of Aizu. He is a full professor with the Department of Computing, The Hong Kong Polytechnic University. His research interests are mainly in the areas of big data, cloud computing and networking, and distributed systems with more than 400 papers published in major conferences and journals. His work was recognized by the 2016 Annual Best of Computing: Notable Books and Articles in Computing in ACM Computing Reviews. He is the recipient of the 2017 IEEE Systems Journal Annual Best Paper Award and other five Best Paper Awards from IEEE/ACM conferences. He was an associate editor of the *IEEE Transactions on Parallel and Distributed Systems* and an IEEE ComSoc distinguished lecturer. He is now on the editorial board of the *IEEE Transactions on Emerging Topics in Computing*, the *IEEE Transactions on Sustainable Computing*, the *IEEE Transactions on Green Communications and Networking*, and the *IEEE Communications*. He also served as General, TPC and Symposium Chair for numerous IEEE conferences. He currently serves as an officer for several IEEE ComSoc Technical Committees and a director in the ComSoc Board of Governors. He is a senior member of the IEEE.

Albert Y. Zomaya is the chair professor of high performance computing & networking in the School of Information Technologies, University of Sydney, and he also serves as the director of the Centre for Distributed and High Performance Computing. He has published more than 600 scientific papers and articles and is author, coauthor or editor of more than 20 books. He is the founding editor-in-chief of the *IEEE Transactions on Sustainable Computing* and serves as an associate editor for more than 20 leading journals. He served as an editor in-chief for the *IEEE Transactions on Computers* (2011-2014). He is the recipient of the IEEE Technical Committee on Parallel Processing Outstanding Service Award (2011), the IEEE Technical Committee on Scalable Computing Medal for Excellence in Scalable Computing (2011), and the IEEE Computer Society Technical Achievement Award (2014). He is a chartered engineer, a fellow of AAAS, IEEE, and IET. His research interests are in the areas of parallel and distributed computing and complex systems. He is a fellow of the IEEE.