This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2809747, IEEE Transactions on Knowledge and Data Engineering

Tensor-based Big Data Management Scheme for Dimensionality Reduction Problem in Smart Grid Systems: SDN Perspective

Devinder Kaur*, Gagangeet Singh Aujla†, *Student Member, IEEE*, Neeraj Kumar‡, *Senior Member, IEEE*, Albert Y Zomaya§, *Fellow, IEEE*, Charith Perera¶, *Member, IEEE*, Rajiv Ranjan‖, *Senior Member, IEEE*

Abstract—Smart grid (SG) is an integration of traditional power grid with advanced information and communication infrastructure for bidirectional energy flow between grid and end users. A huge amount of data is being generated by various smart devices deployed in SG systems. Such a massive data generation from various smart devices in SG systems may generate issues such as-congestion, and available bandwidth on the networking infrastructure deployed between users and the grid. Hence, an efficient data transmission technique is required for providing desired QoS to the end users in this environment. Generally, the data generated by smart devices in SG has high dimensions in the form of multiple heterogeneous attributes, values of which are changed with time. The high dimensions of data may affect the performance of most of the designed solutions in this environment. Most of the existing schemes reported in the literature have complex operations for data dimensionality reduction problem which may deteriorate the performance of any implemented solution for this problem. To address these challenges, in this paper, a tensor-based big data management scheme is proposed for dimensionality reduction problem of big data generated from various smart devices. In the proposed scheme, firstly the Frobenius norm is applied on high-order tensors (used for data representation) to minimize the reconstruction error of the reduced tensors. Then, an empirical probability-based control algorithm is designed to estimate an optimal path to forward the reduced data using software-defined networks (SDN) for minimization of the load and effective bandwidth utilization on the network infrastructure. The proposed scheme minimizes the transmission delay occurred during the movement of the dimensionally reduced data between different nodes. The efficacy of the proposed scheme has been evaluated using extensive simulations carried out on the data traces using 'R' programming and Matlab. The big data traces considered for evaluation consist of more than 2 million entries (2075259) colecetd at 1 minute sampling rate having hetrogenous features such as–voltage, energy, frequency, electric signals, etc. Moreover, a comparative study for different data traces and a real SG testbed is also presented to prove the efficacy of the proposed scheme. The results obtained depict the effectiveness of the proposed scheme with respect to the parameters such as- network delay, accuracy, and throughput.

Index Terms—Big data, Dimensionality reduction, Flow table management, Smart grid, Software-defined networks, Tensors.

1 Introduction

Smart grid (SG) is an intelligent power grid which supports a bidirectional energy flow between users and grid using advanced information and communication technologies (ICT)-based infrastructure. It optimizes user's demand, energy generated, and network availability to provide reliability and efficiency using automated controls, sensors, metering devices, and distributed energy sources. It contains components such as smart meters and sensing devices connected to one another using communication infrastructure. The delivery of various services such as energy, voltage, and frequency regulations to the end users depends upon the reliable, and real-time information about the data flow between users and service providers (grid). For this purpose, a reliable communication infrastructure is required to manage the flow of data and information between sources of data generation and smart meters. Advanced metering infrastructure (AMI) and Phasor measurement units (PMU) are the two main infrastructure units used for acquiring the data generated from different smart devices in SG systems and then pass the collected data to the utility which takes decisions about energy flow. AMIs are the bi-directional units which contain sensing devices, smart meters, control and monitoring systems, and data management units. On the other hand, PMUs are the energy measurement units generally used to measure energy waves and signals [1]. Data acquisition systems are used for sampling or collecting the analog data which is further converted into numeric values using computing technology. Moreover, sensors are the main data acquisition components which convert the physi-

- *D. Kaur is with the Computer Science & Engineering Department, Thapar Institute of Engineering and Technology, Patiala (Punjab), India and also with Lovely Professional University, Phagwara (Punjab), India E-mail: kaurdevinder07@gmail.com and*
- *G. S. Aujla is with the Computer Science & Engineering Department, Thapar Institute of Engineering and Technology, Patiala (Punjab), India and also with the Computer Science & Engineering Department, Chandigarh University, Gharuan (Punjab), India. E-mail: gagi_aujla82@yahoo.com*
- *N. Kumar is with the Computer Science & Engineering Department, Thapar Institute of Engineering and Technology, Patiala (Punjab), India E-mail: neeraj.kumar@thapar.edu*
- *Albert Y. Zomaya is with School of Information Technologies, J12, University of Sydney, NSW 2006, Australia E-mail: albert.zomaya@sydney.edu.au*
- *C. Perera is with Open Lab, NewCastle University, United Kingdom (UK) E-mail: charith.perera@newcastle.ac.uk*
- *R. Ranjan is with Computer Science Department, Newcastle University, United Kingdom (UK) E-mail: raj.ranjan@ncl.ac.uk*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2809747, IEEE Transactions on Knowledge and Data Engineering

cal parameters such as–temperature and voltage to electrical signals in a discrete form. These devices are deployed in different smart communities in large number across the globe for effective power management generated from various distributed energy resources in SG environment. For example, approximately 50,000 smart meters are deployed by US department of energy and Los Angeles department of water and power (LADWP) in the Los Angeles itself. LADWP serves approximately 4.1 million consumers which accounts to nearly 1% of total US power consumption [2]. Another report by energy information administration (EIA), highlights the need of an efficient data management with higher penetration of renewable energy sources (RES) to manage demand and supply optimally [3].

The description of various data generation, transmission, and distribution units in SG is as shown in Fig. 1. Handling the large amount of data generated by smart devices in different time intervals using AMIs and PMUs is one of the biggest challenges in SG systems. The enormous amount of data acquired at the SG level often leads to the problems related to QoS provisioning and demand response management [1]. The data is generated at regular intervals depending upon the deployment of smart devices across different geographical regions. With the advent of smart homes equipped with many smart devices, the frequency of data generation increased many folds which in turn poses challenges of data representation, data storage, and processing at SG level. As most of the smart devices generate data with high sampling rate so handling the volume and velocity of the data need to be done in such a manner so that efficient decisions with respect to demand response can be taken on time [4]. Moreover, analyzing the SG big data may play a vital role in an intelligent power distribution such as–prediction of power patterns, demand response, fault tolerance, and RES management.

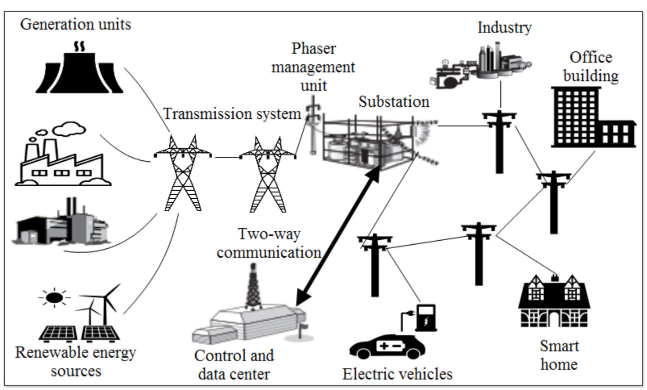


Fig. 1: An overview of the smart grid

1.1 Related Work

The major issue with big data handling is its complexity due to the presence of multidimensional and heterogeneous attributes [5], [6]. Hence, the conversion of SG big data into a simplified structure is required for faster processing. In this direction, Souza *et al.* [7] presented a data compression methodology in smart distribution systems based upon the singular value decomposition (SVD) technique. The methodology presented by authors is based on the lossy data compression method. In lossy compression, the new value of data is bound to lose its originality by a certain value. Therefore, in order to reconstruct the original data without losing valuable information, a trade-off between compression ratio and the reduction rate needs to be maintained. For this purpose, authors in [8] presented a data compression technique using SVD in smart distribution systems. Moreover, Ning *et al.* [9] proposed a data compression technique based upon the wavelet function. This technique compresses the size of noise signals, which in turn affects data transmission. However, in order to represent heterogeneous big data with reduced dimensions, tensor representation is one of the emerging techniques [10]. For example, Kuang *et al.* [11] proposed a tensor-based unified model for big data representation and size reduction. Yang *et al.* [12] introduced a similar technique named as lanczos-based high order SVD algorithm to reduce the dimensionality of unified data tensor model.

After effective storage and representation of big data, another major task is to transmit the reduced data over the underlying SG network efficiently [13]. In this context, authors in [14], [15] reviewed the key aspects of smart metering process with a focus on the type of data generated and techniques required to process it. Authors highlighted that the data processing at network level is a major challenge faced by SG systems. This is due to the generation of data at regular intervals from various smart devices leading to the traffic congestion at the network infrastructure. Similarly, Plaza *et al.* [16] presented the possibility of reception and information broadcasting between smart meters and the grid, in real-time through the cellular network using AMI. Hence, after analyzing the aforementioned proposals, it is evident that efficient data flow over the existing network infrastructure is required for handling the big data generated from various smart devices in SG systems.

To mitigate these challenges, Software-defined network (SDN) has emerged as a flexible platform for efficient traffic flow. Authors in [17], [18] presented various features of SDN such as- network capabilities, interfaces, and programming languages used. They have highlighted that SDN makes the network management tasks easier, due to the decoupling of data plane with the central control plane. Mckeown *et al.* [19] elaborated the use and deployment of the communication protocol of SDN called as OpenFlow. Authors in [20]–[22] highlighted the deployment of centrally controlled SDNs in wireless sensor networks and network operating systems. Kim *et al.* [23] highlighted various benefits of using SDN explicitly in different environments. Due to the logic and flexibility involved, it becomes very easy to reconfigure the network changes dynamically. Similarly, authors in [24] presented an SDN-based communication architecture for microgrid. Moreover, authors in [25], [26] introduced the SDN and cloud related prototypes for the SG communications in order to provide flexible and reliable services to the end users. Cahn *et al.* [27] explored the benefits of integrating SDN in the SG systems. The authors utilized SDN to design a self-managed substation network for SG systems. In an another work, Aujla *et al.* [28] utilized SDN for energy management for sustainable DCs. Moreover, big

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2809747, IEEE Transactions on Knowledge and Data Engineering

data and SDN are interrelated to each other and utilize the properties of each other for mutual benefits. SDN benefits the big data applications by improving the performance of the network in the form of latency and throughput [29]. In this regard, Kuang *et al.* [30] proposed a tensor-based big data approach in SDN for effective QoS provisioning.

1.2 Motivation

After analyzing the aforementioned proposals, it is inferred that a huge amount of big data is being generated by various smart devices in the SG. However, handling this big data in an efficient manner is one of the biggest challenges in SG environment. An efficient data handling and processing of big data at SG systems can lead to a better demand response management, energy consumption prediction, and effective communication among various devices. Various techniques have been analyzed with respect to these issues in the existing proposals [14] - [15]. But, none of the existing proposals have focused on big data analytics in SG systems for an efficient QoS provisioning. Moreover, the existing proposals have not explored any unified model for data representation. In recent times, tensors (apart from vectors and matrices) have been effectively used for representation and management of big data using SDN [11] - [12]. Also, the big data represented by tensors could be reduced to a simpler form by removing the redundant and ambiguous dimensions. Moreover, to ease the burden of data flow on the existing network infrastructure, SDN can play an integral role in processing and forwarding the reduced data in an efficient manner over the SG network infrastructure [17]- [23], [29]- [30]. Hence, there is a need of an unified and intelligent SDN model for big data management in SG systems. Therefore, a tensor-based SDN model for efficient big data management in SG systems has been designed in the proposal.

1.2.1 Motivation examples

For better illustration of the proposed scheme, let us consider an example as shown in Fig. 2. If a data frame of 20 Mbits is transmitted over traditional network channel, then it takes 1 second to reach destination node at a data rate of 20 Mbps (refer Fig. 2 (a)). However, if the data is reduced (16 Mbits) in size omitting all invaluable and unwanted values, then it takes 0.8 seconds to reach to the destination at the same data rate (refer Fig. 2(b)). Hence, it is quite evident, that the size of data has a strong impact on the transmission time. Since, a huge amount of data is transmitted seamlessly in SG systems so if such data is reduced then it may be beneficial for the overall performance of the network.

Now, when the underlying networks follow a dynamic network management scheme, then it can help to achieve better utilization of network resources and thereby can achieve enhanced throughput. For example, if a reduced data frame is transmitted over traditional networks, they may choose the shortest path using traditional network protocols. In such a case, a data rate of 50 Mbps is achieved along with a link utilization of 25% and throughput of 12.5 Mbps (refer Fig. 2(c)). However, if dynamic networks such as SDN are deployed, then a data rate of 55 Mbps is achieved along with a better link utilization of 25.2% and an enhanced throughput of 13.7 Mbps (refer Fig. 2(d)). Hence, it clearly shows that a better throughput, data rate, and link utilization can be achieved by deployment of SDN based network infrastructure for data management in SG systems. The better utilization of link may also help in reducing the energy consumption of network infrastructure.

1.3 Research contributions of this work

Based upon the above discussion, the major contributions of this work are as given below.

1) A tensor-based data management scheme is designed for representation and dimensionality reduction of data acquired from various smart devices in SG systems. Then, the Frobenius norm is applied to optimize the reconstruction error of the reduced tensor.
2) An empirical probability-based control algorithm is designed for estimation of an optimal route to forward the reduced data over SG networks using SDN.
3) The proposed scheme is evaluated using extensive simulations on data traces taken at per-minute sampling rate for 4 years (December 2006 to November 2010) [31], PJM dataset [32], and a real SG test bed.

1.4 Organization

The remaining paper is organized as follows. Section II represents the problem formulation. Section III elaborates the proposed scheme. Section IV elaborates the mathematical case study for tensor representation and dimensionality reduction. The results and discussions are presented in Section V. Finally, Section VI concludes the paper.

NOMENCLATURE	
D_{ac}	Acquired data
D_ϕ	Unstructured data
D_ψ	Semi-structured data
D_ω	Structured data
T	Tensor
a_n	Orders of tensor
x_n	Attributes
T_ϕ	Sub-tensors for unstructured data
T_ψ	Sub-tensors for semi-structured data
T_ω	Sub-tensors for structured data
T_{uni}	Unified tensor
M_i	Matrix of mode-i
U, V	Urinary matrix
S	Diagonal matrix
V^*	Conjugate transpose of urinary matrix V
σ	Singular values
r	Lower rank
e_n	Dimensional attributes of n^{th} order tensor
T_{red}	Reduced core tensor
$\hat{T}_{red}$	Approximated tensor
M	Multi-dimensional array
m, n	Dimensions of array
ε	Reconstruction error
ρ	Reconstruction error ratio
N	Number of nodes
L	Set of links
$c(l)$	Channel capacity
$f(l)$	Traffic flow on link l
N_r	Updated flow table entry
$\hat{\theta}$	Empirical distribution function
n_p	New data packet forwarded to controller
o_p	Older observations in a flow table

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2809747, IEEE Transactions on Knowledge and Data Engineering

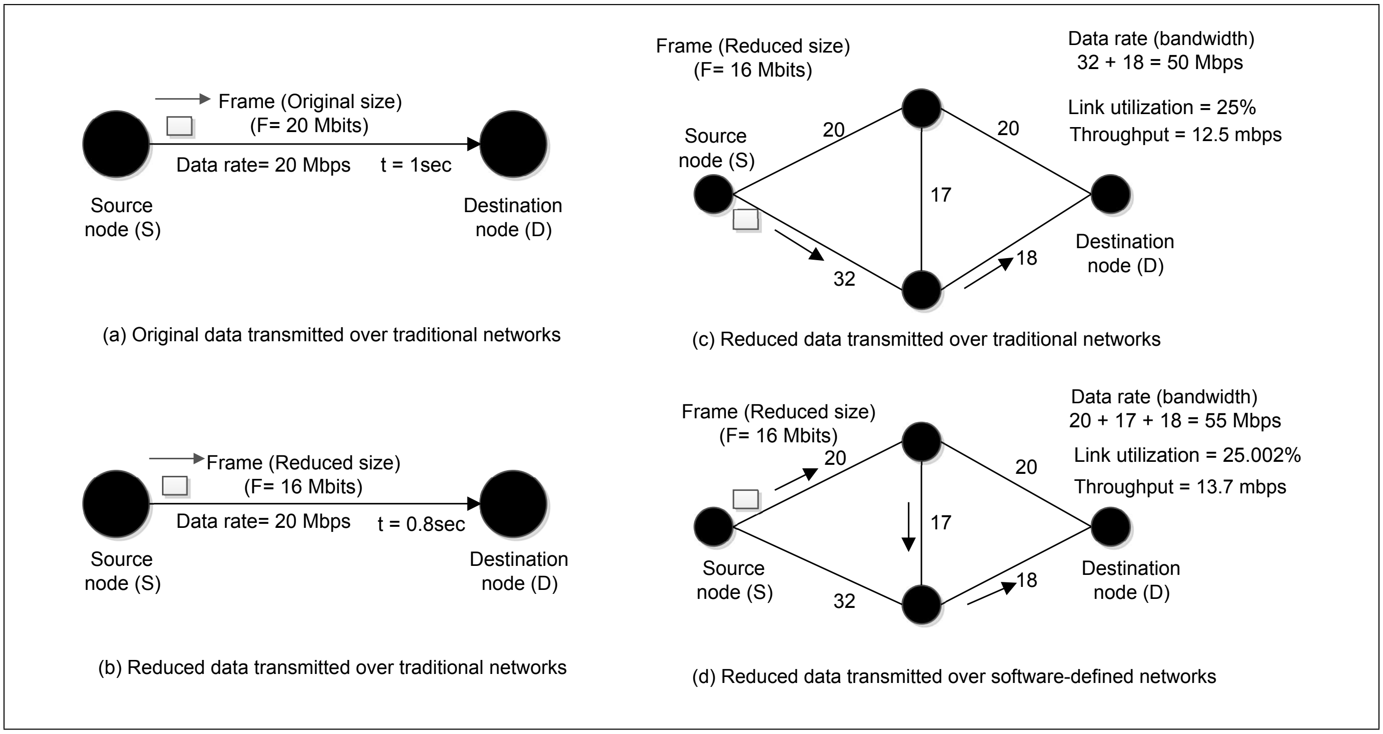


Fig. 2: Tensor-based SDN model

2 Problem Formulation

In this paper, three different types of datasets D_ϕ, D_ψ, and D_ω are used for unstructured, semi-structured and structured data, respectively. These datasets are acquired from various smart devices incorporated in SG systems. The acquired data is represented in the tensor form. Tensors are multi-way arrays which are used to represent the data having multiple characteristics and high dimensions. A tensor T of n-order is represented as follows.

$$T \in R^{a_1 \times a_2 \times a_3 \ldots \times a_n} \tag{1}$$

where, a_1, a_2,.., a_n are the orders of tensor which define the dimensionality of data characteristics.

In order to represent big data as tensors, data with n number of characteristics is represented as a cross product of various characteristics having multiple dimensions. The representation of big data in tensor form is given as follows.

$$E[x_1 \otimes x_2 \otimes x_3 \otimes ..x_n] = R^{a_1 \times a_2 \times a_3 .. \times a_n} \tag{2}$$

Here, $x_1, x_2, ..., x_n$ represent different attributes present in big data (for example, voltage, energy consumption, meter/customer ID, and load can be described as various attributes of SG big data data generated by smart devices).

Each attribute of big data is independent of the other and can be represented as a cross product of each other. Hence, using Eq. (2), the acquired heterogeneous big datasets are converted into their respective tensors as given below.

$$D_\phi \rightarrow T_\phi, D_\psi \rightarrow T_\psi, D_\omega \rightarrow T_\omega \tag{3}$$

where, T_ϕ, T_ψ, and T_ω denotes sub-tensors for unstructured, semi-structured, and structured data, respectively.

In order to reduce the data redundancy and duplicacy, sub-tensors are converted into an unified tensor (T_{uni}) using a *unified data tensorization operation* as a function given below [12].

$$f : (D_\phi \cup D_\psi \cup D_\omega) \rightarrow T_\phi \cup T_\psi \cup T_\omega \tag{4}$$

$$f(x,y,z) = u \tag{5}$$

where x $\in T_\phi, y \in T_\psi, z \in T_\omega$, and u $\in T_{uni}$

The union operator combines the similar characteristics and reduces the redundancy from the acquired big data. However, with the presence of higher dimensionality, the complexity of big data remains high which leads to data inconsistency and data processing problems in big data analytics. To overcome such problems, the unified tensor needs to be transformed into a lower-order tensor having fewer dimensions which can be represented as the reduced tensor. The transformation of a n^{th} order tensor into n number of matrices is known as tensor unfolding or matricization [11]. For a given tensor, $T \in R^{a_1 \times a_2 \times a_3 \ldots \times a_n}$, the equation of unfolding n-order matrix into a mode-i matrix is given as below.

$$T \in R^{a_i \times (a_1 \times a_2 \times a_3 \ldots \times a_{i-1} \times a_{i+1} \ldots \times a_n)} \tag{6}$$

The number of rows and columns of each mode-i matrix are given by Eq. (7) and Eq. (8), respectively.

$$a_i, 1 \leq i \leq n \tag{7}$$

$$\prod_{j=1}^{n} a_j, i \neq j \tag{8}$$

Now, SVD is used to factorize a real or a complex matrix. An unfolded mode-i matrix (M_i) which is to be decomposed using SVD, can be represented as given below.

$$M_i = U_i S_i V_i^* \tag{9}$$

where, U and V are unitary matrices and are orthogonal to each other, S is a diagonal matrix, V^* is the conjugate transpose of the unitary matrix V.

The diagonal matrix S, with non-negative entries has i singular values denoted by σ_i. The singular values (σ_i) in the diagonal matrix S are as given below.

$$S = \begin{bmatrix} \sigma_1 & \cdots & \cdots & \cdots & 0 \\ 0 & \sigma_2 & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & \sigma_i & 0 \end{bmatrix}. \tag{10}$$

$$S = diag(\sigma_1, \sigma_2, ..., \sigma_i, 0, ..., 0) \tag{11}$$

$$where;\ \sigma_1 \geq \sigma_2 \geq ...... \geq \sigma_i > 0 \tag{12}$$

After applying SVD on each mode-*i* matrix, the rank of a singular matrix is approximated to a lower rank r ($r \leq n$). The $(r-k)$ values obtained from the singular matrix are truncated and a low-rank approximation is achieved by applying the SVD incrementally. Rank is approximated based upon the threshold value of 30 percent rank reduction depending upon the target dataset. Then, the reduced tensor is obtained by projecting the orthogonal vectors obtained from the results of truncated SVD, over the initial tensor (T_{uni}). The dimensionality reduction is achieved by obtaining reduced tensor which contains the lesser dimensions, but valuable and core information as present in the initial tensor. The dimensionality of the n^{th} tensor is reduced using n-mode product operation. The n-mode product operation of a tensor (T_{uni}) by a matrix (U) is defined as follows.

$$(T_{uni} \times_n U)_{e_1 e_2 ....e_{k-1} e_k e_{k+1} ....e_n} \tag{13}$$

where, $e_1 e_2 ....e_{k-1} e_k e_{k+1} ....e_n$ are the dimensional attributes of n^{th} order tensor.

In order to calculate the reduced core tensor (T_{red}), n-mode product is used for a n^{th} order tensor as shown below.

$$T_{red} = T_{uni} \times_{x=1}^{n} U_n^T \tag{14}$$

$$T_{red} = T \times_1 U_2^T \times_2 U_3^T ..... \times_n U_n^T \tag{15}$$

Moreover, from this reduced tensor, an approximated tensor ($\hat{T}_{red}$) can be reconstructed as illustrated below.

$$\hat{T}_{red} = T_{red} \times_1 U_1 \times_2 U_2 \times_3 U_3 .... \times_n U_n \tag{16}$$

The approximation can be further optimized using Frobenius-norm on the tensor values obtained after tensor product by a matrix. The Frobenius norm is one of the important matrix norms which finds the size of a multidimensional array M, by taking the square root of the sum of the squares of its elements as given below.

$$||M||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{m} (M_{ij})^2} \tag{17}$$

Frobenius norm on M having two dimensions m, n dimensions, is defined as below.

$$||M||_F = \sqrt{\sum_{i=1}^{min(m,n)} (\sigma_i)^2} \tag{18}$$

The reconstruction error defines the approximation accuracy of the reduced tensor. It occurs due to the approximation of mode-*i* matrices. The reconstruction error for unified tensor and approximated reduced tensor is given as below.

$$\varepsilon = ||T_{uni} - \hat{T}_{red}||_F \tag{19}$$

With an increase in reconstruction error ratio, the accuracy of the core data or reduced tensor decreases. The reconstruction error ratio, ρ can be analyzed using Frobenius-norm of original unified tensor and final reduced tensor and is defined as below.

$$\rho = \left(\frac{||T_{uni} - \hat{T}_{red}||_F}{||T_{uni}||_F} \right) \tag{20}$$

Hence, the main objective function of the proposed scheme is to minimize the reconstruction error and is defined as below.

$$min\,(\rho) \tag{21}$$
$$s.t. \tag{22}$$
$$\rho \in [0, 1] \tag{23}$$
$$T_{uni} > \hat{T}_{red} \tag{24}$$
$$T_{uni} > 0 \tag{25}$$
$$\hat{T}_{red} > 0 \tag{26}$$
$$S_{rt} \propto \frac{1}{\varepsilon} \tag{27}$$
$$T_{rt} \leq B_{ch} \tag{28}$$

where, S_{rt} is sampling rate, T_{rt} is the transmission rate and B_{ch} is bandwidth of the channel.

3 Proposed Scheme

Fig. 3 shows the work flow of the proposed scheme.

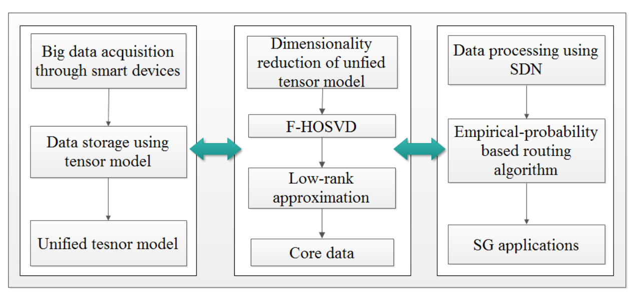


Fig. 3: Workflow of the proposed scheme

3.1 Tensor-based data management scheme

In this scheme, a tensor-based data management scheme is presented to acquire raw data and reduce it to lower dimensionality thereby optimizing the reconstruction error. The acquisition of big data in SG environment involves various challenges such as–missing values, inconsistency, duplicate or redundant values, heterogeneity, different formats, sampling rate, etc. However, the proposed scheme handle these challenges in an effective manner. In this regard, algorithm 1 is designed and described as below.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2809747, IEEE Transactions on Knowledge and Data Engineering

The acquired data (D_{ac}) is sorted in structured, semi-structured, and unstructured data (line 1-2). After sorting the data, each type of data is converted into its corresponding sub-tensors (T_ϕ, T_ψ, and T_ω) using Eq. (2) (line 3). Then, all the sub-tensors are combined together to form a unified tensor (T_{uni}) using Eq. (4). Now, the unified tensor (T_{uni}) is unfolded into n matrices using Eq. (6) (line 5). Then, all the unfolded matrices are decomposed using SVD. The matrices are decomposed into a combination of unitary matrix U, conjugate transpose of unitary matrix V (V^*), and a diagonal matrix S (line 6-9). Now, lowest rank approximation is applied to keep r largest singular values and replacing other values by zero (line 10-13). Then, $\hat{M}_i$ which is used to obtain the approximated decomposed values ($U_i, \hat{S}$,V_i^*) is calculated using Eq. (9) (line 14-17). The n-mode product is applied to the left orthonormal column vectors with the initial tensor to obtain the reduced tensor, T_{red} using Eq. (15) (line 18-20). After this, $\hat{T}_{red}$ is calculated using Eq. (16). The Frobenius norm for minimizing the difference between original and approximated reduced tensor is applied on the reconstruction error ratio (ρ) to optimize the result. If (ρ) is less than the threshold value of error ratio (ρ_{th}), then the reduced tensor (T_{red}) is sent to the destination. Otherwise, repeat the process till it satisfies the acceptable error ratio (line 21-29).

Algorithm 1 Tensor-based data management algorithm

Input: D_{ac}, Acquired raw data
Output: T_{uni}, T_{red}

```
1:  Acquire D_ac from various sources
2:  Sort into D_φ, D_ψ, and D_ω
3:  D_φ → T_φ, D_ψ → T_ψ, D_ω → T_ω
4:  Unify T_φ, T_ψ and T_ω using Eq. (4)
5:  Unfold T_uni into n matrices using Eq. (6)
6:  for (i=1; i ≤ n; i++) do
7:      Apply SVD(M_i)
8:      Obtain U_i, S_i and V_i^*
9:      Extract σ_i from S_i
10:     Calculate rank(S_i) = n
11:     while do(1 < r ≤ n)
12:         Obtain (Ŝ) by pruning the smallest σ_i
13:         Obtain rank(Ŝ)
14:         Reconstruct (M̂_i) using Eq. (9).
15:         Extract new Û_i, Ŝ_i, andV̂_i^*
16:     end while
17:     Store the left truncated orthonormal vectors, Û_i.
18:     Perform n-mode product of Û_i with T_uni.
19:     Calculate T_red using Eq.(15).
20: end for
21: Reconstruct T̂_red using Eq.(16).
22: Apply Frobenius-norm on T̂_red using Eq.(17).
23: Compute ε, using Eq. (19)
24: Obtain ρ, using Eq. (20)
25: if (ρ < ρ_th) then
26:     Send T_red to destination.
27: else
28:     Recalculate T_red to satisfy ρ.
29: end if
```

4 SDN-BASED CONTROL SCHEME

In this section, an emerging software-centric networking paradigm called SDN is used in the proposed scheme to provide dynamic network management in SG systems. SDN is an open and programmable platform which controls the network in an intelligent and dynamic way through well-decoupled planes. It provides abstraction of underlying infrastructure from network applications, which makes it easy to manage and reconfigure according to the dynamic changes int the network configuration [27]. The growing rates of big data traffic at SG systems could be effectively handled using scalability and efficiency of SDN. Hence, the integration of big data technologies such as tensor models with SDN can led to an extensible and efficient service provisioning to the end users. In this context, a tensor-based SDN model is designed in the proposed scheme using three planes; (1) data plane, (2) control plane, and (3) application plane as shown in Fig. 4.

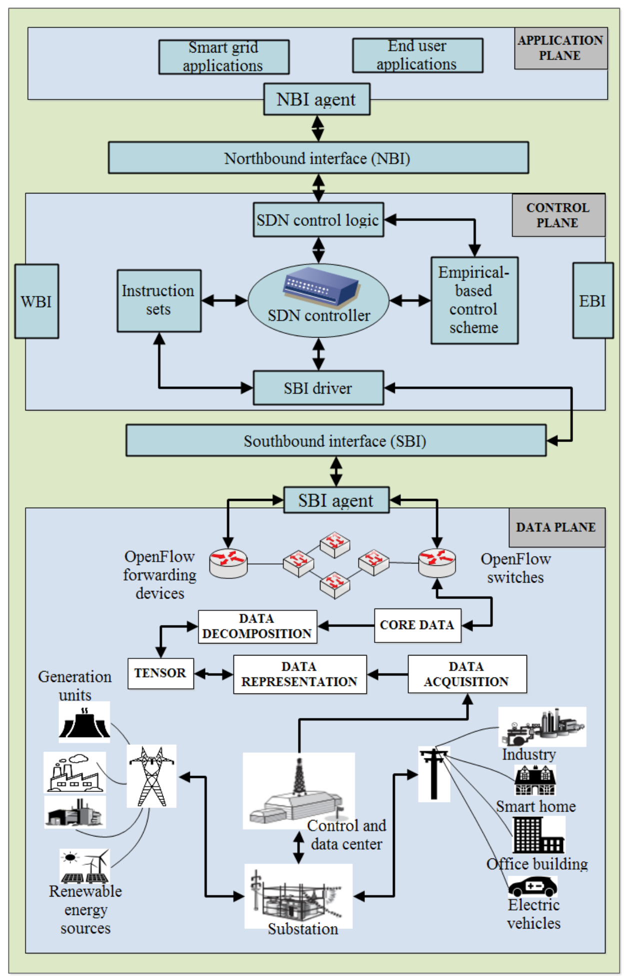


Fig. 4: Tensor-based SDN model

In this model, data plane mainly consists of network devices such as- switches and routers. Data is acquired from various devices such as- appliances in smart homes, and electric vehicles (EVs). This plane use open flow protocol (OFP) as a communication standard to forward the gathered data to the upper plane [27]. The acquired data is decom-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2809747, IEEE Transactions on Knowledge and Data Engineering

posed into a reduced tensor of smaller rank and size at the server located at the control plane. With the help of control algorithms, core data is processed in an efficient manner. For this purpose, an empirical probability-based control scheme is designed to estimate the optimal route for transmission of reduced data over SG networks. Finally, the application plane provides various services to different users and SG.

4.1 Flow Table Management

The data plane consist of forwarding nodes (FNs) such as openflow switches, routers, and gateways. At the control plane, SDN controller is responsible for taking the forwarding decisions for FNs. These decisions are configured into FNs using southbound interface. A set of flow tables and group tables that are linked to each other by a pipeline resides in the FNs [33]–[37].

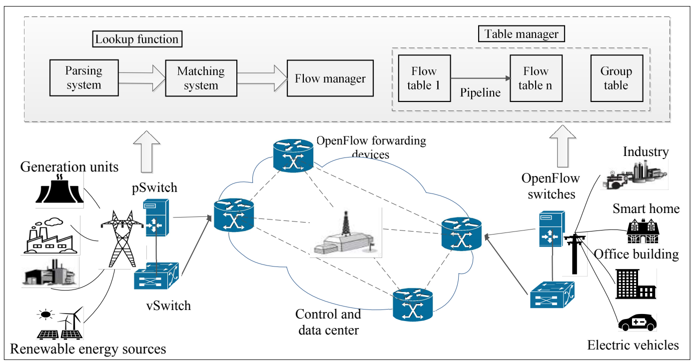


Fig. 5: Data Plane Management

A flow table comprising of different fields such as–entry id, instructions, priority, action, port number, etc follows the instruction set provided by SDN controller. Fig. 5 shows the flow matching process in SDN model. The major steps for controller-switch communications are given as below.

- **Step 1**: Initially, the incoming packet is analyzed by the parsing system to decide how it can be processed. This involves three main steps, (1) header identification, (2) field extraction, and (3) field buffer [38]. The output is communicated to the lookup header after analyzing the packet.
- **Step 2**: After receiving the output, the lookup function initiates with the ingress port and finishes with the egress port. Generally, two types of lookup methods: 1) exact matching, and 2) wild-card matching are used. Exact Matching uses a hash function for setting the exact position of a particular item. Wild-card Matching is a complex partial string matching for multiple table lookup, designed to match the header field of an entry.
- **Step 3**: Once the OF-switches receives the packet, the matching system initiate the task of deciding the routing decision. If a suitable match is found in the table, then the corresponding action is performed. In such a case, the packet is forwarded to the concerned port and the OF-switch updates its first counter, i.e., matched counter. On the contrary, if the matching fails, then the packet is send back to the controller. In such a case, the OF-switch updates its second counter, i.e., mismatch counter field. Now, the controller rebuilds a new flow rule and inserts it at the OF-switch. A flow-driven rule caching algorithm (FDRCA) [39] is used to replace the entries in the flow table. FDRCA is a policy-based algorithm which handles the limited cache size constraints and unpredictable flows by pre-fetching and special replacement strategy.
- **Step 4**: A flow table pipeline is used to connect all the tables (table *0* to table *n*). The group tables (contains entries that are concerned with the variety of actions that affects one or more flows) and meter table (contains entries associated with the performance related information) are also available.
- **Step 6**: A table manager uses a counter to record the number of packets sent to the controller.
- **Step 7**: Finally, after matching the packet header field successfully, the outgoing packet is directed through egress switch port on the basis of the action set.

Table I shows the flow table entries maintained at each OF-switch. Flow table entries such as–source IP, destination IP, priority, port number ingress port, and action are some of the most important for taking decision about data transfer between different entities. However, this may vary dynamically according to the situation and requirements of incoming flow. The list of various flow table entries are given as below.

- **Table no.**: The number of a flow table, i.e, its relative position in the flow table pipeline.
- **Entry id**: A *unique id* (primary key) is assigned to each entry in the flow table of an OF-switch.
- **Priority**: Importance of each entry in the flow table.
- **Ingress port**: A physical/virtual port where the incoming packet arrives.
- **VLAN id**: It contains 13-bits virtual LAN *id* and 3-bits VLAN type.
- **Ethernet address**: It contains 48-bit MAC address for each flow entry. It can be an exact address or a wildcard matching.
- **IPv4/IPv6 address**: It consists of a 32-bit IPv4 address or an 128-bit IPv6 address.
- **TCP/UDP port number**: It contains a 16-bit TCP/UDP source and destination port number.
- **Action**: The instructions to be followed once packet matches with an associated rule are given here.
- **Counter**: It is assigned for various attributes such as–byte counter, packet counter, flag, flow duration, and number of dropped packets.
- **Timeouts**: It contains the expiry duration of a flow rule. This can be of two types; 1) Hard timeout, and 2) Idle timeout.
- **Cookie**: It consists of flow statistics, deletion and modification entries that are managed by the controller [38].

4.2 Empirical probability-based control scheme (EPCS)

As T_{red} needs to be transmitted over the underlined network through optimal paths. Since, the incoming traffic flow

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2809747, IEEE Transactions on Knowledge and Data Engineering

TABLE 1: Flow table entries of an typical OF-switch

Table no	Entry ID	Priority	Ingress Port	VLAN id	eth dst	eth src	eth type	ipv4 dst	ipv4 src	tcp dst	tcp src	Action	Counter
0	30	099	1	210	B	A	0x0800,.	103.42.0.0/16	192.168.0.0/24	tcp dst:80	tcp src:83	encap, fwd:[2]	11
0	170	097	5	154	D	A	0x0800,.	103.20.0.0/16	192.168.1.0/22	tcp dst:81	tcp src:83	send to controller	12
1	605	090	4	153	C	A	0x0800,.	103.0.0.0/13	192.168.6.0/24	tcp dst:82	tcp src:83	drop	11
1	112	085	10	44	A	B	0x0800,.	104.31.0.0/16	192.168.8.0/24	tcp dst:80	tcp src:0	send to controller	12
2	192	077	14	140	B	C	0x0800,.	101.27.9.0/24	192.168.9.0/22	tcp dst:21	tcp src:0	encap, fwd:[2,4]	10
2	154	052	3	112	C	D	0x0800,.	103.0.0.0/13	192.168.3.0/24	tcp dst:20	tcp src: 25	drop	11
.	-	-	-	-	-	-	-	-	-	-	-	-	-
.	-	-	-	-	-	-	-	-	-	-	-	-	-
n	175	007	13	74	E	C	0x0800,.	100.24.3.0/24	192.168.81.0/24	tcp dst:23	tcp src: 123	tunnel to controller	15

is scheduled for different applications in SG environment. So, different QoS requirements exist for each incoming traffic flow. Hence, low latency queuing (LLQ) model is best suited for the incoming traffic flow [40], [41]. So, laying the foundation on traditional SDN routing algorithm an empirical-probability based scheme is proposed in this section to recommend routing paths with maximum likelihood for scheduling or reduced data. It would tend to maximize the channel utilization and minimize the latency. This scheme tends to shape the traffic coming from various smart devices at real time, with respect to the resource availability and given QoS constraints after the data reduction has been performed. SDN switches interact with controller which estimates the optimal routes. For this purpose, the link characteristics such as-bandwidth, channel capacity, latency, load, etc. are considered for selecting optimal destination nodes. In order to keep the track of various network related updates like path priorities, a flow table is maintained with the help of programs or logic applied by the controller at the control plane.

In this regard, a routing scheme for the SDN controller is presented in order to forward the reduced data through optimal paths with lower latencies and higher QoS. In this scheme, all the FNs can be visualized as SDN FNs and non-SDN FNs. Data which passes through at least one SDN FN comes under controllable flow and which does not passes through any SDN FN, is considered as uncontrollable flow [42]. Now, consider a network *Z(N,L)*, having *N* number of nodes and *L* links, *c(l)* denotes the channel capacity and *f(l)* refers to the traffic flow on link *l*. Now the flow table already has old entries in it. Through our proposed logic, we tend to update the flow table entry to a new one (N_r) The older flow table entries are taken as the observation set which are to be fed to the estimator at SDN controller. It further predicts or estimates the optimal path (N_p).

Empirical Probability is an estimation of the occurrence of an event, happening in an actual environment. We can estimate N_r using probabilistic approach. The empirical distribution function (estimator) $\hat{\theta}$, can be given as.

$$\hat{\theta} = \frac{n_p}{o_p} \tag{29}$$

where, n_p denotes new data packet forwarded to the controller, and o_p denotes the older observations in a flow table.

The optimality of the scheme can be checked using the *mean-square error (MSE)* of the estimator. MSE of the estimator $\hat{\theta}$, is defined as the function of the new routes to be predicted as shown below.

$$MSE(\hat{\theta}) = f(n_r) \tag{30}$$

$$E[(\hat{\theta} - \theta)^2] = E[(g(n_r) - \theta)^2)] \tag{31}$$

The algorithm for the proposed scheme is given as below.

Algorithm 2 Empirical probability-based control algorithm

Input: Z(N,L), n_p.
Output: Maximize channel utilization, u.

1: Split the Z(N,L) into SDN and non-SDN FNs.
2: Forward n_p to the SDN controller.
3: $\forall$ destination d $\in$ N, apply OSPF on non-SDN FNs.
4: Obtain o_p.
5: **while** SDN FNs $\subset$ N **do**
6: Feed the o_p to $\hat{\theta}$.
7: Compute $\hat{\theta}$ using Eq. 29.
8: Obtain N_p.
9: Update flow table, N_r.
10: **while** MSE($\hat{\theta}$) $\leq THR_{crlb}$ **do**
11: max u.
12: **end while**
13: **end while**

In the proposed algorithm, the network Z(N,l) is divided into type of forwarding nodes called SDN FNS and non-SDN FNs (line 1). Now, the data packets (n_p) are forwarded to the controller (line 2). Further, using open shortest path first algorithm, entries are updated in the flow table and o_p is maintained (line 3-4). After updating flow table, the empirical estimator is applied on SDN-FNs to estimate the new path for the data at the controller. The new flow table is then updated with new estimated values (line 5-9). The accuracy of the predicted routes is checked using MSE of the estimator. The MSE obtained is compared with the threshold value (THR_{crlb}). If the value of MSE is less than or equal to the THR_{crlb}, then the channel utilization is maximized (line 10-11).

The value of THR_{crlb} is computed using CramrRao lower bound (CRLB) method [43], [44]. CRLB states that the variance of any unbiased estimator is at least as high as the inverse of the Fisher information ($I(\theta)$). If the estimator reaches the CRLB, it is said to be efficient. The condition for MSE using CRLB is given as below.

$$E[(\hat{\theta} - \theta)^2] \geq \frac{1}{I(\theta)} \tag{32}$$

Here, the $I(\theta)$ is given as below.

$$I(\theta) = -E\Big[\frac{\partial^2}{\partial\theta^2} log f(X;\theta)\Big] \tag{33}$$

5 Mathematical Case Study

The following section represents the exemplar case study for tensor-based data representation and dimensionality

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2809747, IEEE Transactions on Knowledge and Data Engineering

reduction. The high-dimensional big data is represented using tensors. Fig. 6, shows the visualization of a three-order tensor $R^{4\times5\times3}$ having 4, 5, and 3 instances at each order, respectively.

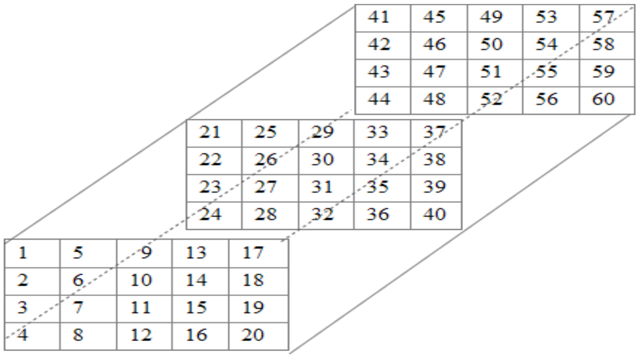


Fig. 6: A visualization of a three-order tensor, $R^{2\times3\times4}$

An n-order tensor can be unfolded into n different matrices through the process of matricization. The transformation of a high-order tensor into lower order matrices is known as tensor unfolding or matricization. For a given tensor $R^{4\times5\times3}$, the process of unfolding into matrices can be done using Eq. (6). Fig. 7 shows the tensor $R^{4\times5\times3}$ that has been unfolded into three different matrices named as *M1*, *M2*, and *M3*, respectively. The row and column number of each matrix is calculated using Eqs. (7) and (8), respectively. Now, M1 has been unfolded by taking first order as row number and the product of rest of the orders contribute to column number. For example, in M1, there are four number of rows and fifteen number of columns. In a similar manner, others matrices (M2 and M3) can be expanded.

$$M1 = \begin{matrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 \end{matrix}$$

$$M2 = \begin{matrix} 1 & 6 & 11 & 16 & 21 & 26 & 31 & 36 & 41 & 46 & 51 & 56 \\ 2 & 7 & 12 & 17 & 22 & 27 & 32 & 37 & 42 & 47 & 52 & 57 \\ 3 & 8 & 13 & 18 & 23 & 28 & 33 & 38 & 43 & 48 & 53 & 58 \\ 4 & 9 & 14 & 19 & 24 & 29 & 34 & 39 & 44 & 49 & 54 & 59 \\ 5 & 10 & 15 & 20 & 25 & 30 & 35 & 40 & 45 & 50 & 55 & 60 \end{matrix}$$

$$M3 = \begin{matrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 & 25 & 28 & 31 & 34 & 37 & 40 & 43 & 46 & 49 & 52 & 55 & 58 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 & 26 & 29 & 32 & 35 & 38 & 41 & 44 & 47 & 50 & 53 & 56 & 59 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 27 & 30 & 33 & 36 & 39 & 42 & 45 & 48 & 51 & 54 & 57 & 60 \end{matrix}$$

Fig. 7: Unfolding of tensor $R^{2\times3\times4}$ into three matrices

After the matricization of a tensor into matrices, SVD is applied on on each matrix truncated in order to obtain singular values as shown in the Fig. 8. Each matrix gets decomposed further into three matrices, i.e., two orthonormal matrices (U_i and V_i*) and a diagonal matrix (S_i). The diagonal matrix contains singular values in descending order, i.e., $a_1 \geq a_2 \geq ....a_i \geq 0$.

After applying truncated SVD on each matrix (M_i), the null values in the diagonal matrix can be pruned and top largest values having rank r can be retained. Table 2 shows various mathematical results obtained for each matrix. Column II shows the number of singular values obtained and next three columns shows the top 3 largest singular values optimized after truncating the null values. By spanning the singular value space with orthonormal vectors of decomposed matrix, an approximate matrix with with rank (r) is obtained. After this, in order to calculate the reduced tensor T_{red}, n-mode product is used for n-order tensors as shown in the Eqs. (15) and (16). The dimensionality of the given tensor is reduced by incrementally applying n-mode product on initial tenor with left orthonormal space. Now, the reduced tensor can be approximated by optimizing the error reduction ratio using Frobenius norm.

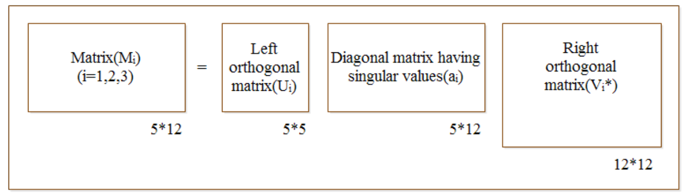


Fig. 8: Singular Values Decomposition

TABLE 2: Mathematical results

Matrix no.	i	σ_1	σ_2	σ_3
M1	5	271.626	5.391	0.301
M2	4	271.646	4.267	0.000
M3	3	271.662	3.119	0.005

6 Results and Discussions

In this section, the proposed tensor-based SDN model for management of big data generated by SG devices using proposed scheme is evaluated using data traces for individual household electric power consumption [31]. The dataset consist of 2075259 measurements gathered with a one-minute sampling rate for about 4 years (December 2006 to November 2010). The dataset contains some missing values along with various sub-metering and electrical quantity values [31]. The results obtained after extensive simulation are compared with HOSVD scheme using 'R' programming and Matlab. The objective of proposed scheme is to minimize the reconstruction error ratio between unified tensor and reduced tensor using Frobenius norm. To evaluate the proposed scheme, a network topology is designed in Mininet network emulator [45].

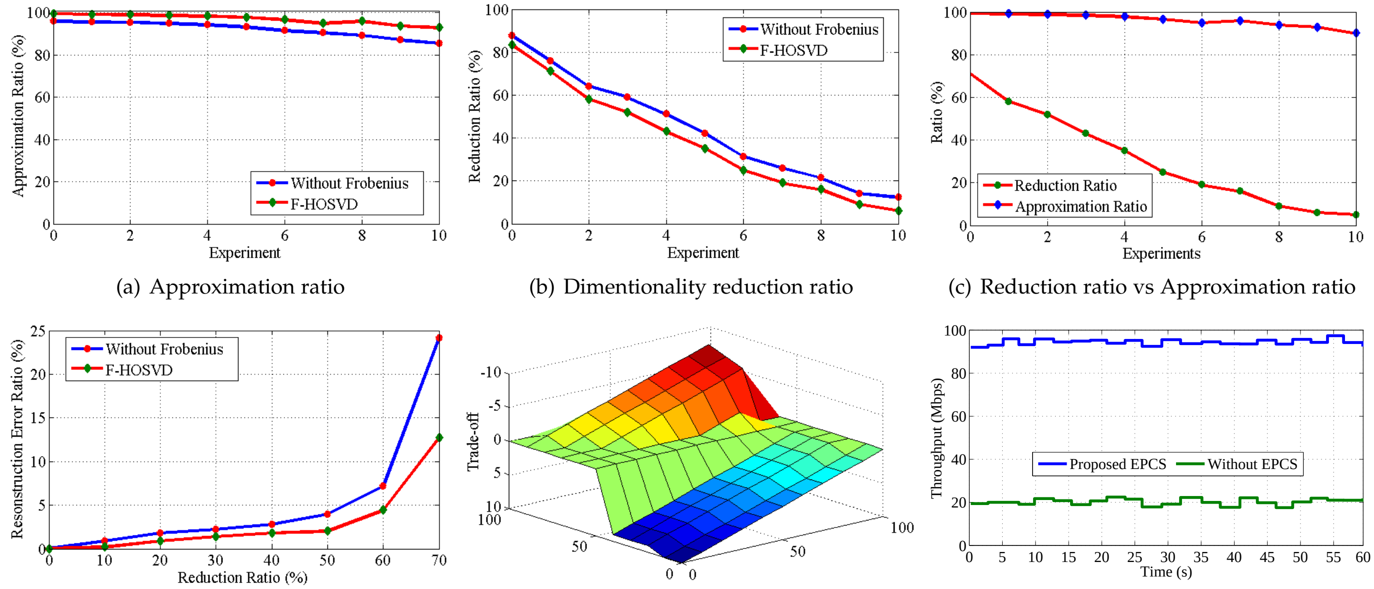
6.1 Evaluation parameters

The proposed scheme has been evaluated using following parameters.

- **Dimensionality reduction ratio** (λ) is the ratio of the non-zero values of the reduced tensor and orthonormal vectors to the non-zero values of the initial tensor. The λ for the initial tensor T_{uni} is given by as below.

$$\lambda = \frac{nz(\hat{T}_{red}) + \sum_{i=1}^{n} nz(U_i)}{nz(T_{uni})} \tag{34}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2809747, IEEE Transactions on Knowledge and Data Engineering



(a) Approximation ratio
(b) Dimentionality reduction ratio
(c) Reduction ratio vs Approximation ratio
(d) Recontruction error ratio
(e) Trade-off
(f) Throughput

Fig. 9: Evaluation results for the proposed scheme

- **Approximation accuracy** is the trade-off between the reconstruction error ratio (ρ) and dimensionality reduction ratio (λ) and are inversely proportional to each other.
- **Delay** (d) is the latency at a specific router and comprise of processing delay (d_{pr}), queuing delay (d_q), transmission delay (d_t), and propagation delay (d_{pg}) [45].

$$d = d_{pr} + d_q + d_t + d_{pg} \tag{35}$$

- **Network throughput** is the rate of successful delivery of message over a certain communication channel. It can also be called as the maximum rate at which data can be processed.

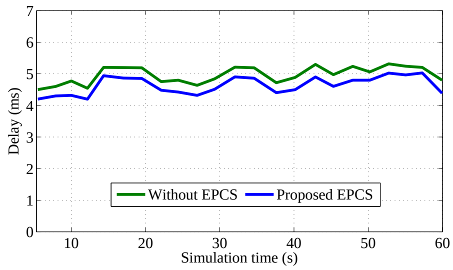
6.2 Evaluation Results

The data acquired is converted into sub-tensors (using Eq. (2)). Then, the sub-tensors are combined to form a unified tensor (using Eq. (4)). The unified tensor is obtained by applying unified data tensorization operation on the sub-tensors. The unified tensor is reduced to obtain a lower order tensor using F-HOSVD technique. The unified tensor combines the sub-tensors to remove all the ambiguities, redundancies to obtain a simplified combined tensor.

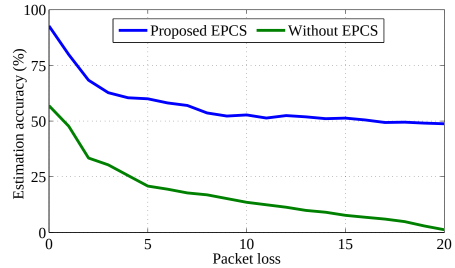
The reduced tensor is an approximation of the original data which contains all the valuable information. The results obtained show that the approximation ratio obtained using the proposed scheme is more as compared to the existing technique. The approximation ratio decreases from 99.5% to 89.9% with respect to a decrease in reduction ratio from 83.4% to 5%. Hence, it is clear that the nearly 90% originality of the data is maintained even after reduction up to 5%. Fig. 9(a) shows the approximation ratio obtained after performing experiments on the original data. Further, the reduction ratio obtained using the proposed scheme is shown in Fig. 9(b). It shows that the original tensor is reduced to a higher extent as compared to simple HOSVD. Hence, it clearly depicts that the data is reduced to a higher ratio while maintaining originality. The comparison of reduction ratio with respect to the approximation ratio is shown in Fig. 9(c). The above results are obtained using Frobenius norm which is applied incrementally on the singular matrix to achieve a lower rank matrix. The above conclusion is further supported by the results obtained for the reconstruction error ratio. The reconstruction error ratio obtained for the experiments using the proposed scheme is shown in Fig. 9(d). The results depict that reconstruction error ratio is lower as compared to the existing technique. Therefore, the proposed scheme shows higher level of originality while achieving lower reconstruction error. The above results are achieved using Frobenius norm which is applied incrementally on tensors after n-mode product. Hence, the overall objective of minimizing the reconstruction error ratio is achieved using the proposed scheme. The trade-off between reduction ratio and error ratio is shown in Fig. 9(e).

Once the data acquired from various SG devices has been reduced into core data then it has to be processed and transmitted over SG networks using SDN infrastructure. For this purpose, an empirical probability-based control scheme has been designed to estimate an optimal path for the reduced data. After evaluation of the proposed scheme, it is evident that the all the performance metrics shows a suitable growth. Fig. 9(f) shows the throughput achieved for the proposed route estimation scheme. The results obtained shows a higher throughput is achieved by using empirical probability-based control scheme. Also, the delay incurred for transmitting the data to the destination is lower with respect to standard SDN routing scheme. Fig. 10(a) shows the delay incurred while transmitting the reduced data over SG networks using the proposed route estimator along with SDN. The proposed scheme is evaluated for the estimation accuracy with respect to packet loss. The results obtained

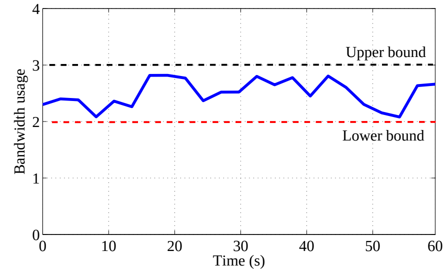
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2809747, IEEE Transactions on Knowledge and Data Engineering



(a) Delay



(b) Estimation accuracy



(c) Bandwidth usage

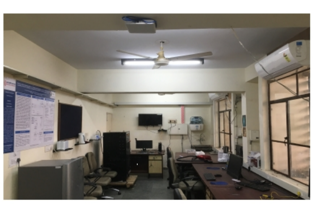
Fig. 10: Evaluation results for the proposed scheme

are evident that the optimal route estimation accuracy for the proposed scheme is better that the standard open flow scheme. It is also evident that the accuracy drop is less for the proposed scheme with respect to increase in packet loss. Fig. 10(b) shows the achieved estimation accuracy with respect to packet loss. In this regard, Table II shows the values of RMSE obtained for estimated routes.

TABLE 3: RMSE values for EPCS

Samples	Data sent (Mbits)	RMSE	Bandwidth range (Mbps)
S1	8	.06025	10-50
S2	16	.041472	100-150
S3	32	.18245	150-200
S4	64	.139355	200-250
S5	128	.1409	250-300

Finally, the proposed scheme is evaluated with respect to the bandwidth usage. The results clearly depict that the stability of bandwidth usage is maintained within the upper bound and lower bound. Hence, it shows that no congestion or bandwidth over/under utilization occurs. Fig. 10(c) shows the bandwidth usage for the proposed scheme for route estimation. This strongly shows that the available bandwidth is optimally utilized by the control scheme for transmitting reduced data over SG networks. The above discussed evaluation results depict the effectiveness and efficiency of the proposed scheme with respect to various performance metrics.



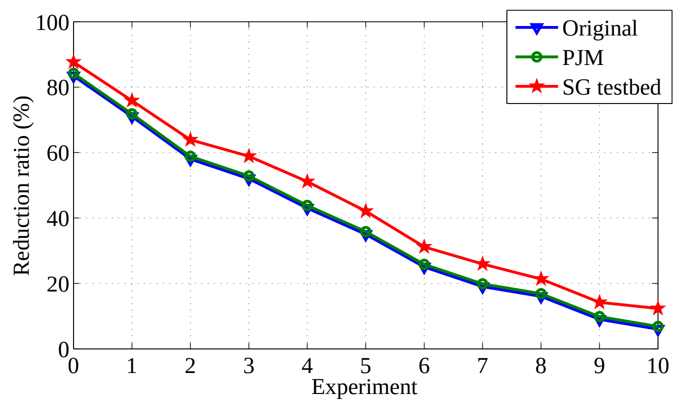
(a) Experimental SG testbed

(b) Grid simulator and prototype

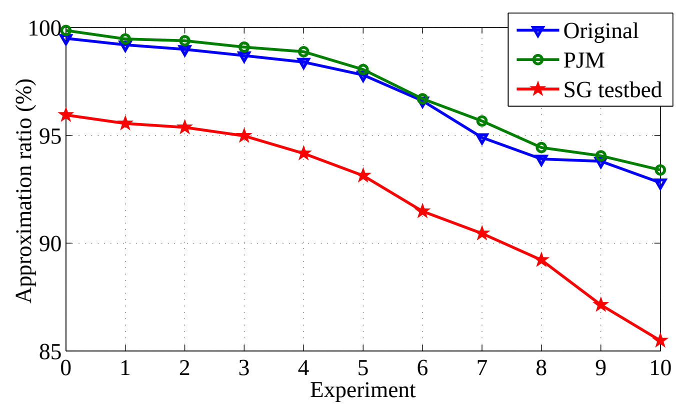
Fig. 11: SG testbed setup

6.3 Comparative study for different data traces

In this subsection, a comparative study for three different data traces namely 1) original [31], 2) PJM [32], and 3) real SG test bed is performed. Table 4 shows the comparative analysis of these data traces with respect to different sampling rates and parameters. For this purpose, a data traces for a real SG test bed are collected at 1 second sampling rate for 6 months. The SG testbed including various smart home appliances, a prototype with STM8S microcontroller, SP1ML RF transceiver, IC (MAX 232), and relays are deployed. Figs. 11(a) and 11(b) shows the experimental SG testbed along with controller and grid simulator.



(a) Dimentionality reduction ratio



(b) Approximation ratio

Fig. 12: Comparative analysis of different data traces

After simulations, the results obtained for different datasets are shown in Table 4. The impact of sampling rate is clearly visible on the results obtained. The higher is the sampling rate, the higher is efficiency. Fig. 12(a) shows the dimensionality reduction ratio for all three data traces. The original and PJM datasets shows almost similar results for reduction ratio. However, the results for SG testbed are lower than others. Similarly, the approximation ratio for original and PJM datasets shows similar trend. However, the approximation ratio for SG testbed dataset is lower in contrast to the other two. Fig. 12(b) shows the approximation ratio for all three datasets. From the above comparison, it is evident that the proposed scheme performs efficiently for different datasets. The average reconstruction error ratio obtained for original (4.43 percent), PJM (4.98), and SG testbed (6.12 percent) datasets shows the effectiveness of the proposed scheme.

TABLE 4: Comparative analysis

Data Traces	S_{rt}	RR (%)	AR (%)	ρ (%)
Original [31]	1 minute	38.23	96.33	4.43
PJM [32]	1 hour	39.11	96.93	4.98
SG testbed	1 minute	32.43	92.89	6.12

RR: Reduction ration, AR: Approximation ratio

7 Conclusion

In this paper, a tensor-based SDN model for dimensionality reduction problem for big data acquired from various SG devices is proposed. For this purpose, a F-HOSVD algorithm is designed. The purpose of the proposed scheme is to represent the bulk data generated by SG devices in a tensor form. After tensor representation, the sub-tensors are combined to form a unified tensor. Finally, the proposed algorithm for dimensionality reduction is applied on the unified tensor to reduce it. The proposed scheme is validated using data traces for individual household energy consumption. The results obtained show that the proposed scheme achieves higher dimensionality reduction while maintaining a high ratio of originality. Also, the reconstruction error ratio of the data is minimal as compared to the existing techniques. Moreover, a comparative study for different data traces and a real SG testbed is also presented to prove the effectiveness of the proposed scheme. Finally, the proposed empirical probability-based control scheme for SDN is used to estimate path for forwarding the reduced data. The results show that the estimated path show low latency and high throughput. Moreover, the proposed scheme maintains a high route estimation accuracy with respect to increase in packet loss. Finally, the bandwidth utilization remains stable and the proposed scheme avoid any congestion or under utilization of bandwidth. Hence, the overall results obtained for proposed schemes related to data management, dimensionality reduction, and route estimation shows better performance than existing schemes.

References

[1] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "Smart grid technologies: communication technologies and standards," *IEEE transactions on Industrial informatics*, vol. 7, no. 4, pp. 529–539, Nov 2011.

[2] Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou, and V. Prasanna, "Cloud-based software platform for big data analytics in smart grids," *Computing in Science Engineering*, vol. 15, no. 4, pp. 38–47, Jul 2013.

[3] "Annual energy outlook 2010 with projections to 2035," *Energy Information Administration*, Apr 2010, (Accesed on: Apr 2017). [Online]. Available: http://www20.iadb.org/intal/catalogo/pe/2010/05499.pdf

[4] Y. Demchenko, C. de Laat, and P. Membrey, "Defining architecture components of the big data ecosystem," in *International Conference on Collaboration Technologies and Systems*, 2014, pp. 104–112.

[5] R. Ranjan, L. Wang, A. Y. Zomaya, J. Tao, P. P. Jayaraman, and D. Georgakopoulos, "Advances in methods and techniques for processing streaming big data in datacentre clouds," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 2, pp. 262–265, 2016.

[6] L. Zhao, L. Chen, R. Ranjan, K.-K. R. Choo, and J. He, "Geographical information system parallelization for spatial big data processing: a review," *Cluster Computing*, vol. 19, no. 1, pp. 139–152, 2016.

[7] J. C. S. de Souza, T. M. L. Assis, and B. C. Pal, "Data compression in smart distribution systems via singular value decomposition," *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 275–284, 2015.

[8] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[9] J. Ning, J. Wang, W. Gao, and C. Liu, "A wavelet-based data compression technique for smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 212–218, Mar 2011.

[10] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[11] L. Kuang, F. Hao, L. T. Yang, M. Lin, C. Luo, and G. Min, "A tensor-based approach for big data representation and dimensionality reduction," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 280–291, Sep 2014.

[12] L. T. Yang, L. Kuang, J. Chen, F. Hao, and C. Luo, "A holistic approach to distributed dimensionality reduction of big data," *IEEE Transactions on Cloud Computing*, 2015, doi: 10.1109/TCC.2015.2449855.

[13] M. U. S. Khan, O. Khalid, Y. Huang, R. Ranjan, F. Zhang, J. Cao, B. Veeravalli, S. U. Khan, K. Li, and A. Y. Zomaya, "Macroserv: A route recommendation service for large-scale evacuations," *IEEE Transactions on Services Computing*, vol. 10, no. 4, pp. 589–602, 2017.

[14] D. Alahakoon and X. Yu, "Smart electricity meter data intelligence for future energy systems: A survey," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 1, pp. 425–436, Feb 2016.

[15] D. Zhou, J. Guo, Y. Zhang, J. Chai, H. Liu, Y. Liu, C. Huang, X. Gui, and Y. Liu, "Distributed data analytics platform for wide-area synchrophasor measurement systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2397–2405, Sep 2016.

[16] F. P. Plaza and E. I. Ortega, "State of art, meter data management system using compressed sensing for AMI based on wavelet." *IEEE Latin America Transactions*, vol. 13, no. 12, pp. 3774–3780, 2015.

[17] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[18] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[19] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[20] K. Jeong, J. Kim, and Y.-T. Kim, "QoS-aware network operating system for software defined networking with generalized openflows," in *IEEE Network Operations and Management Symposium*, 2012, pp. 1167–1174.

[21] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: towards an operating system for networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 3, pp. 105–110, 2008.

[22] T. Luo, H.-P. Tan, and T. Q. Quek, "Sensor openflow: Enabling software-defined wireless sensor networks," *IEEE Communications Letters*, vol. 16, no. 11, pp. 1896–1899, 2012.

[23] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2809747, IEEE Transactions on Knowledge and Data Engineering

[24] L. Ren, Y. Qin, B. Wang, P. Zhang, P. B. Luh, and R. Jin, "Enabling resilient microgrid through programmable network," *IEEE Transactions on Smart Grid*, 2016, doi: 10.1109/TSG.2016.2589903.

[25] K. Akkaya, A. S. Uluagac, and A. Aydeger, "Software defined networking for wireless local networks in smart grid," in *IEEE 40th Local Computer Networks Conference Workshops*, 2015, pp. 826–831.

[26] R. Narayanan, K. Xu, K.-C. Wang, and G. K. Venayagamoorthy, "An information infrastructure framework for smart grids leveraging SDN and cloud," in *Clemson University Power Systems Conference*, 2016, pp. 1–7.

[27] A. Cahn, J. Hoyos, M. Hulse, and E. Keller, "Software-defined energy communication networks: From substation automation to future smart grids," in *IEEE International Conference on Smart Grid Communications*, 2013, pp. 558–563.

[28] G. S. Aujla, A. Jindal, N. Kumar, and M. Singh, "SDN-based data center energy management system using RES and electric vehicles," in *IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[29] L. Cui, F. R. Yu, and Q. Yan, "When big data meets software-defined networking: SDN for big data and big data for SDN," *IEEE Network*, vol. 30, no. 1, pp. 58–65, Jan 2016.

[30] L. Kuang, L. T. Yang, X. Wang, P. Wang, and Y. Zhao, "A tensor-based big data model for QoS improvement in software defined networks," *IEEE Network*, vol. 30, no. 1, pp. 30–35, 2016.

[31] M. Lichman, "UCI machine learning repository," 2013, (Accesed on: Apr 2017). [Online]. Available: http://archive.ics.uci.edu/ml

[32] *Metered Load Data*, (Accessed on: Nov 2017). [Online]. Available: http://www.pjm.com/markets-and-operations/ops-analysis/historical-load-data.aspx

[33] G. S. Aujla and N. Kumar, "MEnSuS: An efficient scheme for energy management with sustainability of cloud data centers in edge-cloud environment," *Future Generation Computer Systems*, 2017, doi: 10.1016/j.future.2017.09.066.

[34] G. S. Aujla, R. Chaudhary, N. Kumar, J. J. Rodrigues, and A. Vinel, "Data offloading in 5g-enabled software-defined vehicular networks: A stackelberg-game-based approach," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 100–108, 2017.

[35] G. S. Aujla, N. Kumar, A. Y. Zomaya, and R. Rajan, "Optimal decision making for big data processing at edge-cloud environment: An SDN perspective," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 778–789, 2018.

[36] G. S. Aujla and N. Kumar, "SDN-based energy management scheme for sustainability of data centers: An analysis on renewable energy sources and electric vehicles participation," *Journal of Parallel and Distributed Computing*, 2017, doi: 10.1016/j.jpdc.2017.07.002.

[37] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. P. C. Rodrigues, and M. Guizani, "Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 44–51, Feb 2018.

[38] R. Chaudhary, G. S. Aujla, N. Kumar, and J. J. P. C. Rodrigues, "Optimized big data management across multi-cloud data centers: Software-defined-network-based analysis," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 118–126, Feb 2018.

[39] H. Li, S. Guo, C. Wu, and J. Li, "Fdrc: Flow-driven rule caching optimization in software defined networking," in *IEEE International Conference on Communications (ICC)*, 2015, pp. 5777–5782.

[40] F. Li, J. Cao, X. Wang, and Y. Sun, "A sdn-based qos guaranteed technique for cloud applications," *IEEE Access*, vol. 5, pp. 21 229 – 21 241, 2017.

[41] *Low Latency Queueing - LLQ. Class-Based Weighted Fair Queueing*, (Accesed on: Apr 2017). [Online]. Available: http://www.cisco.com/c/en/us/td/docs/ios/120s/feature/guide/fsllq26.html.

[42] S. Agarwal, M. Kodialam, and T. Lakshman, "Traffic engineering in software defined networks," in *IEEE INFOCOM*, 2013, pp. 2211–2219.

[43] A. Van den Bos, "A cramér-rao lower bound for complex parameters," *IEEE Transactions on Signal Processing*, vol. 42, no. 10, p. 2859, Oct 1994.

[44] B. Hamschin and M. Grabbe, "An approximate cramer-rao lower bound for multiple lfmcw signals," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 3, pp. 1365 – 1374, Jun 2017.

[45] F. Ongaro, "Enhancing quality of service in software-defined networks," Ph.D. dissertation, Alma Mater Studiorum-University of Bologna, 2014.

Devinder Kaur is working as an Assistant Professor with the Computer Science and Engineering Department at Lovely Professional University, Phagwara, Punjab, India. Prior to this, she was pursuing her M.Tech degree in Computer Science and Engineering Department from Thapar Institute of Engineering and Technology (Deemed to be University), Patiala, Punjab, India. She received the B.Tech degree in Information Technology from Punjab Technical University, Jalandhar, Punjab, India in 2013. Her research interests include data analytics, machine learning, big data, and smart grid.

Gagangeet Singh Aujla is working as an Associate Professor in Computer Science and Engineering Department at Chandigarh University, Gharuan, Mohali, Punjab, India. Prior to tis, he was working towards his Ph.D. from Thapar Institute of Engineering and Technology (Deemed to be University), Patiala, Punjab, India. He received the B.Tech degree in Computer Science and Engineering from Punjab Technical University, Jalandhar, Punjab, India, in 2003, and the M.Tech degree in Computer Science and Engineering from Punjab Technical University, Jalandhar, Punjab, India, in 2013. He has many research contributions in the area of smart grid, cloud computing, edge computing, vehicular networks, software defined networks, security and cryptography. Some of his research findings are published in top cited journals such as IEEE TII, IEEE TCC, IEEE Communication Magazine, IEEE CE Magazine, FGCS, and JPDC. He is member of the IEEE, CSI, and ISTE.

Neeraj Kumar received his Ph.D. in CSE from Shri Mata Vaishno Devi University, Katra (J & K), India, and was a postdoctoral research fellow in Coventry University, Coventry, UK. He is working as an Associate Professor in the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology (Deemed to be University), Patiala (Pb.), India. He has published more than 200 technical research papers in leading journals and conferences from IEEE, Elsevier, Springer, John Wiley etc. Some of his research findings are published in top cited journals such as IEEE TIE, IEEE TDSC, IEEE TITS, IEEE TCE, IEEE TII, IEEE TVT, IEEE ITS, IEEE Netw., IEEE Comm., IEEE WC, IEEE IoTJ, IEEE SJ, FGCS, JNCA, and ComCom. He has guided many research scholars leading to Ph.D. and M.E./M.Tech. His research is supported by funding from UGC, DST, CSIR, and TCS. He is an Associate Technical Editor of *IEEE Communication Magazine* and an Associate Editor of *IJCS*, Wiley, *JNCA*, Elsevier, and *Security & Communication*, Wiley. He is senior member of the IEEE.

Albert Y. Zomaya is currently the Chair Professor of High Performance Computing & Networking in the School of Information Technologies, The University of Sydney. He is also the Director of the Centre for Distributed and High Performance Computing which was established in late 2009. Professor Zomaya published more than 500 scientific papers & articles and is author, co-author/ editor of more than 20 books. He is currently Editor in Chief of the IEEE Transactions on Sustainable Computing. He served as the Editor in Chief of the IEEE Transactions on Computers (2011-2014). Currently, Professor Zomaya serves as an associate editor for 22 leading journals, such as, the ACM Computing Surveys, IEEE Transactions on Computational Social Systems, IEEE Transactions on Cloud Computing, and Journal of Parallel and Distributed Computing. He delivered more than 150 keynote addresses, invited seminars, and media briefings and has been actively involved, in a variety of capacities, in the organization of more than 600 national and international conferences. Professor Zomaya 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 (UK). Professor Zomayas research interests are in the areas of parallel and distributed computing and complex systems

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2809747, IEEE Transactions on Knowledge and Data Engineering

Charith Perera received the B.Sc. (Hons) degree in computer science from Staffordshire University, Stoke-on-Trent, U.K. in 2009 and the Masters degree in business administration from the University of Wales, Cardiff, U.K. in 2012. He received the Ph.D. degree in computer science with The Australian National University, Canberra, Australia. He is currently with The Open University, UK. He was with the Information Engineering Laboratory, ICT Centre, CSIRO, Australia, where he is involved in the EU Open source blueprint for large scale self-organizing cloud environments for IoT applications (FP7 OpenIoT) Project. His research interests include Internet of Things, pervasive and ubiquitous computing with a focus on sensor networks, middleware, context-aware computing, mobile computing, and semantic technologies. Dr. Perera is a member of the Association for Computing Machinery

Rajiv Ranjan is a Reader (equivalent to non-distinguished full professor in the north American system) in Computing Science at Newcastle University, United Kingdom. Before moving to Newcastle University, he was Julius Fellow (2013-2015), Senior Research Scientist and Project Leader in the Digital Productivity and Services Flagship of Commonwealth Scientific and Industrial Research Organization (CSIRO C Australian Governments Premier Research Agency). Prior to that he was a Senior Research Associate (Lecturer level B) in the School of Computer Science and Engineering, University of New South Wales (UNSW). Dr. Ranjan has a PhD (2009) from the department of Computer Science and Software Engineering, the University of Melbourne.