Networking for Big Data: A Survey

Shui Yu, *Senior Member, IEEE*, Meng Liu, *Student Member, IEEE*,
Wanchun Dou, *Member, IEEE*, Xiting Liu, and Sanming Zhou

***Abstract*—Complementary to the fancy big data applications, networking for big data is an indispensable supporting platform for these applications in practice. This emerging research branch has gained extensive attention from both academia and industry in recent years. In this new territory, researchers are facing many unprecedented theoretical and practical challenges. We are therefore motivated to solicit the latest works in this area, aiming to pave a comprehensive and solid starting ground for interested readers. We first clarify the definition of networking for big data based on the cross disciplinary nature and integrated needs of the domain. Second, we present the current understanding of big data from different levels, including its formation, networking features, mathematical representations, and the networking technologies. Third, we discuss the challenges and opportunities from various perspectives in this hopeful field. We further summarize the lessons we learned based on the survey. We humbly hope this paper will shed light for forthcoming researchers to further explore the uncharted part of this promising land.**

***Index Terms*—Big data, networking, mathematical representation, heterogeneity, dynamic network.**

I. Introduction

WE ARE at the doorstep of the age of big data today. With the rapid development of information and communication technology, our society has exhibited an entirely different style. More and more people are enjoying the values extracted from historical and real-time data sets. The typical examples are weather forecast, stock investment, intelligent medical diagnosis, social interaction, and so on. However, in the past decade, the volume of data has increased sharply, which is currently categorized as "big data". According to a report from International Business Machine (IBM), 2.5 quintillion bytes of data are created every day, and 90 percent of the data in the world today were produced within the past two years [1]. For example, the Web is an extraordinary large data sets, we are now using search engines (e.g., Google or Yahoo) and other software tools (e.g., FTP, dropbox) to interact with this gant data set. These big data sets possess rich information and knowledge, which significantly advance various sectors of our society, such as medicine, health-care, business, and so on.

The data science community was the early explorers in big data research [2], [3], and big data is one of the most popular topics in their community due to the significant commercial or political motivations [4], [5]. The world leading researchers pointed out that the appearance of big data is the main engine for the development of data mining and machine learning [6]. We have also seen many successful cases from big data based mining and learning.

However, the fancy mining and learning based big data applications have to depend on the efficient and effective support from the underneath layers in the age of big data. In general, the traditional mining algorithms have two implicit conditions: 1) Locality. All data are stored at the local venue with the computing facility; and 2) Homogeneity. All the data for mining or learning should be homogeneous. However, big data features large-volume, heterogeneous, autonomous sources with distributed and decentralized control, and seeks to explore complex and evolving relationships among data [7]. Namely, the aforementioned two conditions of the current mining or learning algorithms can hardly be satisfied in the forthcoming real big data applications. The data science community had realized the problem decades ago, and some efforts have been invested in heterogeneous data sources [8], [9]. However, to the best of our knowledge, these studies are mainly confined within the data mining or machine learning frameworks in the non big data circumstance.

Complementary to big data mining, researchers from different sectors of computer science are extensively conducting researches to make big data applications feasible and practical. We name these researches as *networking for big data* due to their supporting role in the whole big data picture. It is certain that networking for big data is dispensable for the real forthcoming big data applications. In the real applications of big data in the near future, a big data application usually involve the data sources from different geographically distributed data centers; data fetching among data centers for a given job; job scheduling and synchronization at a global level for a big task;

Today, networking for big data has become an emerging hot topic in networking related communities. There are plenty

Manuscript received March 3, 2016; revised August 4, 2016; accepted September 8, 2016. Date of publication September 19, 2016; date of current version February 22, 2017. This paper is partially supported by the National Science Foundation of China under Grant No. 91318301, No.61672276, and No.61379041; the Key Research and Development Project of Jiangsu Province under Grant No. BE2015154 and BE2016120; and the Collaborative Innovation Center of Novel Software Technology. *(Corresponding author: Wanchun Dou.)*

S. Yu is with the School of IT, Deakin University, Melbourne, VIC 3125, Australia (e-mail: shui.yu@deakin.edu.au).

M. Liu and W. Dou are with the State Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China (e-mail: mengliunju@gmail.com; douwc@nju.edu.cn).

X. Liu is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: lxt670@126.com).

S. Zhou is with the School of Mathematics and Statistics, University of Melbourne, Parkville, VIC 3010, Australia (e-mail: smzhou@ms.unimelb.edu.au).

Digital Object Identifier 10.1109/COMST.2016.2610963

of papers [10], [11], books [12], [13], surveys [14], [15], and tutorials [16] related to big data networking. To the best of our knowledge, the clear concept of networking for big data appeared as a special issue of IEEE Network in 2014 [17], [18], which is followed by several similar special issues in different journals [19], [20], and tracks on the top and flagship conferences, such as IEEE INFOCOM, ICC, and Globecom. We have witnessed the significant development in this emerging field driven by applications with big and bigger data volume, such as the emergence of data center networks, clouds, software-defined networks, and the corresponding software systems, such as MapReduce [21] and OpenFlow [22]. But these new computing platforms are far from perfect to move the hurdles in front of big data applications for pervasive usage in practice. The emergence of big data introduces many unprecedented problems, which can not be served by the existing computer science tools and theories. In terms of networking for big data, we have far more questions than answers. Motivated by the imminent needs and fast developing of networking for big data, we are encouraged to conduct this survey.

Networking for big data is a cross discipline domain. As big data study is still in her infant stage, the research of big data networking interconnects with the other features of big data, such as big data formation, mathematical properties, storage distribution, upper layer application expectation and demands. Therefore, it is necessary to investigate networking for big data in the global environment of big data. As a result, we extend our survey to the related parts of networking for big data, aiming to present a relatively complete picture centered at the networking perspective.

First of all, we have to deeply understand big data to serve it better in terms of networking. Up to now, our understanding of big data is shallow in many aspects. In terms of representation, big data itself can be treated as an abstract network of the interconnected nodes, and researchers generally adopt a graph or matrix to describe networks. In the big data case, there will be a super big graph (or matrix) with hundreds of thousands of nodes and millions of edges. It is challenging to decompose or compress such super big graphs [23]. Moreover, it is awkward to describe the dynamical features of big data networks using the traditional static methods like graph theory.

Secondly, we need to theoretically investigate networking technologies that can be employed for big data applications. We have some achievement in complex networks in the past two decades, such as the small world model [24], the scale-free model [25], the complex network model [26], and the random graph model [27]. However, big data applications raise various open issues with regard to networking to be addressed. For example, many data centers perform big and complex tasks. From a network perspective, these tasks typically compromise multiple flows, which traverse different parts of the network at potentially different times [28]. This new problem is called coflow scheduling, while the traditional scheduling algorithms almost all conduct their study at the single flow case [29]. Researchers have identified a lot of these new demands in big data applications, and a lot more to be uncovered in the years to come.

This paper is the first comprehensive survey on networking for big data to the best of our knowledge. In order to give a better understanding of big data, we provide an overview on its formation process and intrinsic features. As all networking technologies require feasible representations of data, we then summarize the existing works on mathematical representation of big data. More specifically, we elaborate a taxonomy of representations which are able to make big data processable by networking technologies. After that, we categorize and compare a number of relevant research works on networking technologies that can be employed for big data. We then identify open issues raised from networking for big data, and present the exciting opportunities in the emerging research branch. We aim to pave a solid foundation for various communities for further investigation of this promising topic.

The remainder of this paper is organized as follows. In Section II, we give a definition for big data and characterize "networking" for subsequent discussion. We also give a detailed introduction on big data, including its formation and nature. In Section III, we summarize the mathematical representation of big data. In Section IV, we outline the current networking technologies which can serve big data applications. In Section V, we discuss some open issues and feasible research directions. Finally, we summarize this paper in Section VI.

II. Understanding Big Data

In this section, we aim to provide an overview of big data, including the existing works on its definition, formation process, and network side natures. The purpose of this section is to present a foundation for the following study.

A. *Definition of Big Data*

In recent years, the amount of data in our world has been increasing explosively, and analyzing large data sets becomes a key basis underpinning new waves of productivity growth, innovation, and economical surplus [30]. Gartner's 2013 Hype Cycle for Emerging Technologies [31] indicated that big data has reached the peak of inflated expectations, and become one of the most popular techniques.

As a matter of fact, big data concept was defined in year 2000. Diebold [32] presented a paper titled "Big Data" Dynamic Factor Models for Macroeconomic Measurement and Forecasting" in the Eighth World Congress of the Econometric Society. In the paper, they stated "Recently, much good science, whether physical, biological, or social, has been forced to confront and has often benefited from the big data phenomenon." In 2001, Doug Laney, an analyst with the Meta Group, proposed the concept of "3Vs" in his research note titled "3D Data Management: Controlling Data Volume, Velocity, and Variety" [33]. Now, the "3Vs" have become the generally-accepted three defining dimensions of big data, although the term itself does not appear in Laney's note. Following Laney's idea, IDC presented the concept of "4Vs" of big data as Volume, Velocity, Variety, and Value [34]. Here the fourth feature means that values extracted from big data.

In another perspective, IBM believed that the "4Vs" should be: Volume, Velocity, Variety, and Veracity [35]. In 2008, the top journal *Nature* published a special issue on "big data", which examined what big data sets mean for contemporary science [36], [37]. Another top journal *Science* also published a special issue on "Dealing with Data" in 2011, which discussed the "big data" problems in scientific research [38]. A comprehensive review of the definitions of big data can be found in [39].

In this survey, we refer to "big data" as *data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and data process within a "tolerable elapsed time"* [7]. To give an intuitive impression, here we give several examples. The first presidential debate between President Barack Obama and Governor Mitt Romney on 4 October 2012, triggered more than 10 million tweets within 2 hours [40]. Another example is Flickr, a public picture sharing site, which received 1.8 million photos per day, on average, from February to March 2012 [41]. Assuming the size of each photo is 2 megabytes (MB), this requires 3.6 terabytes (TB) storage every single day [7], which raises significant pressure on the storage techniques to existing tools.

B. Formation of Big Data

Human related data sets reflect the interactions of people, and they are excellent and challenging subjects for big data communities. A comprehensive studying of the formation procedure of big data sets will advance our understanding of big data. As a consequence, it is necessary and important to analyze the fundamental data dissemination properties. To the best of our knowledge, however, few research has been done on this topic.

Wang *et al.* [42] argued that the mobile opportunistic networking is a promising technology for big data applications. In mobile opportunistic networks, data dissemination relies on the opportunistic contacts among device carriers. Since currently most devices are carried by human beings, they pointed out that the challenges of analyzing data formation are mainly caused by human mobility and social features. A review of the state-of-the-art analytical results on data dissemination properties of mobile opportunistic networks can be found in the paper. The authors investigated the probability distribution of spreading data to a given region, and analyzed the bounds of the probability at which the data arrives in the objective region within a given time interval. Then they studied the potential of mobile opportunistic networks for big data applications. Ma *et al*. [43] stated that the mobile crowd sensing leveraged pervasive mobile devices to collect big sensory data for large scale Internet of Things applications [44]. The mobile crowd sensing networks not only rely on the opportunistic contacts among device carriers for data transmission, but also leverages human mobility to opportunistically collect sensory data. The authors investigated the opportunistic characteristics of human mobility from the perspectives of both sensing and transmission, and also introduced the state-of-the-art data dissemination law by considering the spatial-temporal correlation among sensory data.

Most of the study on big data formation process are based on epidemic theory. In big data applications, what we usually have is snapshots of the studied objects, such as user connection data sets of Facebook at independent time points. In order to study how a community (structure) formed, we need to traceback to the very early stage of the community, and identify the seed(s) of the community based on the limited available data sets. Community formation depends on message propagation (e.g., through emails or invitations). Epidemic theory is a dominant tool used by researchers to identify the formation process from a network perspective. Various epidemic models have been proposed for source identification, message propagation, community formation, and so on. A comprehensive introduction of networks and epidemic models can be found in [45]. The authors described a variety of methods to present a generic understanding of the relationships between network structure and disease dynamics. They also reviewed some approximation techniques that have been utilized to elucidate the relationship.

The emergent infectious diseases, such as the SARS epidemic in 2003, H1N1 influenza pandemic in 2009 are typical application scenarios for big data. The geographic spread of epidemics are complex, network-driven dynamic processes. In order to timely develop effective containment and mitigation strategies against the diseases, researcher expect to predict their time course, and locate their origin [46]. However, the combined multiscale nature and intrinsic heterogeneity of the underlying networks make it difficult to develop an intuitive understanding of the dissemination processes. By replacing conventional geographic distance with a probabilistically motivated effective distance, the authors showed that complex spatiotemporal patterns can be reduced to surprisingly simple, homogeneous wave propagation patterns. The proposed approach is able to identify the spatial origin of spreading processes and successfully applied to the worldwide 2009 H1N1 influenza pandemic and 2003 SARS epidemic. Even if epidemiological parameters are unknown, it is able to predict relative arrival times of viruses.

The epidemic theory are extensively utilized in the cybersecurity field, which also plays an important role in the big data communities. Zou *et al.* [47] proposed effective algorithms for early detection of the presence of computer viruses, and the corresponding monitoring system. Based on the epidemic model and the observation data from the monitoring system, they successfully detected virus's propagation at its early stage in real-time using the idea of "detecting the trend, not the rate". The proposed scheme can effectively predict the overall vulnerable population size, and correct the bias in the observed number of infected hosts. Simulation results confirm that it can detect the presence of a worm when it infects only 1% to 2% of the vulnerable computers on the Internet.

Malware is pervasive in networks, and poses a critical threat to network security. However, researchers have very limited understanding of malware behavior in networks to date. Yu *et al.* [48] investigated how malware propagate in networks from a global perspective. The authors established a rigorous two layer epidemic model for malware propagation from network to network. Based on the model, the analysis indicated that the distribution of a given malware follows exponential distribution, power law distribution with a short exponential tail, and power law distribution at its early, late and final stages, respectively. The theoretical results are also verified by extensive experiments through two real-world global scale malware data sets.

In summary, literatures about formation of big data sets are very limited up to now. More techniques are expected to be employed to analyze the formation process. Even for the epidemic theory, it suffers a serious accuracy problem in large scale networks [47] and should be modified, improved to meet the needs of big data applications.

C. Features of Big Data From Networking Perspective

Following the definition of big data, we review the intrinsic characteristics of big data in this section. Different from existing works describing "3Vs" or "4Vs", we investigate the essential features related to networking aspects of big data.

1) Distributed Networking: The "distributed networking" feature of big data falls in three-fold. Firstly, big data are usually collected and stored in a distributed way. Autonomous data sources with distributed and decentralized controls are the main characteristics of big data applications [7]. In traditional applications, data are collected over the Internet or the Web, where each server collects a certain information by itself. Every server functions independently without necessarily rely on the others. While for the current big data applications, such as Facebook, Google, Twitter, a large number of distributed server farms are deployed all over the world to provide elastic services and quick responses for local requests. Moreover, the Internet of Things [44] often rely on the distributed networking paradigm to collect sensory data for numerous big data applications. Various types of sensor networks, such as the traditional wireless sensor networks for collecting scalar data [49], the multimedia or video sensor networks for collecting streaming data [50], and the mobile crowd sensing networks for collecting city scale dynamic data [43], [51], always require distributed data storage and processing capabilities.

For big data storage, the GFS (Google File System) [52] is the most widely adopted mechanism for distributed file systems. GFS is designed to run on clusters up to thousands of commodity machines, and could cope with disk, machine, or network fault. Facilitating the development of applications based on GFS, the file system provides a programming interface aimed at abstracting these distribution and management aspects.

Secondly, big data are processed through distributed and parallel ways. Up to date, clouds are the major computing platform for big data. In general, cloud data centers consist of a network of geo-distributed commodity servers providing virtualized computing services [53]. User access services of cloud data centers through allocated virtual machines (VMs). The fundamental technology "virtualization" guarantees that cloud services are isolated at the granularity of VM. But it should be noted that big data applications are generally executed across multiple VMs with a decentralized control pattern. Moreover, cloud execution processes are transparent to users, tasks are assigned to both local and remote VMs to satisfy certain performance requirements (e.g., latency optimal) under resource constraints (e.g., availability) [54]. Among the existing processing models for big data, MapReduce [21] is the most widely used platform. In its application logic, MapReduce is inspired by map and reduce operations, which both run in distributed and parallel fashion.

Thirdly, big data applications are running on top of underlying distributed networks. The recent progress on SDN (Software-Defined Network) further confirms the "distributed networking" nature of big data applications. SDN is a new networking paradigm in which the forwarding hardware is decoupled from control plane [55]. As the emerging applications and services become increasingly complex (e.g., big data applications), SDN promises to dramatically simplify network management and enable innovation and evolution. In SDN protocol like OpenFlow [22], the networking intelligence is logically centralized and physically distributed.

2) Heterogeneous Data Representation: We are now living in an interconnected world. Due to the "distributed networking" nature, most of data are interconnected or interact with each other, forming large scale, sophisticated networks [56]. In most of the current research on network science, social and information networks are usually assumed to be homogeneous, where nodes are objects of the same entity type (e.g., person), and links are relationships of the same type (e.g., friendship). However, big data networks are heterogeneous, where nodes and relations are of different types. On one hand, different information collectors prefer their own schema or protocols for data recording. Namely, there are various representation forms for a certain data instance. Considering a social network of human beings, a person can be represented using various demographic information, such as age, gender, and so on. On the other hand, numerous instances and different applications also lead to diverse data representations. For example, the representation of nodes in a health care network varies according to the instance type, such as patients, doctors, and diseases. While for a given person, images and videos are used to represent him if he is doing a X-ray examination or CT scan.

The "heterogenous data representation" feature introduces many new problems in networking technologies for big data. Imagining a user is trying to search a health care network, then how to aggregate data from all sources is an essential issue if each health practitioner has his own data representations. Given the enormous size and the complexity of the health care network, a user is often only interested in a small portion of the nodes and links most relevant to his queries. As objects are connected and inter-dependent on each other, how to conduct intelligent query and semantic search is a challenge. Moreover, different applications require different concept hierarchies and ontological structures to summarize the networks [56], and multiple pieces of semantic information in heterogenous networks are interwined due to the multiple nodes and links. In order to satisfy the delay requirement (within a "tolerable elapsed time"), the online analytical processing of a heterogenous network is also nontrivial. We will discuss these open issues brought by this nature further in Section V.

III. Mathematical Representation of Big Data

For networking of big data applications, an essential issue is how to represent data with unified mathematical models for further mining and analysis tasks. Different representations can entangle and hide more or less the different explanatory factors of variation behind the data [57]. Consequently, it is necessary to inspect feasible and practical representations of big data.

Intuitively, we should represent big data sets using graphs, then the existing theories and tools for graphs can be applied. However, the data sets in big data cases are usually dynamic, therefore, dynamical representations of big data are absolutely necessary in our study. In order to take advantage of computing power, such as super computers, we need to transform graphs into matrix for computing purpose. As a result, it is highly demanded to explore the field of matrix representation of big data. However, in big data case, the graphs or matrices are extremely big, high dimensional, and complex, making traditional techniques hard to be directly applied. Therefore, compressed sensing and the tensor technologies are necessary to make big graphs smaller, and a complex graphs simpler until we can handle with the given computing facility today.

In this section, we will survey the aforementioned aspects of big data mathematical representation.

A. Graph Representation of Big Data

Graphs provide a powerful primitive for modeling data in a variety of applications. Nodes in graphs usually represent real world objects, and edges indicate relationships between objects. Examples of data modeled as graphs include social networks, biological networks, and dynamic network traffic graphs. In big data applications, graphs are very large, with thousands even millions of nodes and edges. It is almost impossible to understand the information encoded in large graphs by mere visual inspection. In order to make the graph of big data processable using existing tools, techniques such as graph spectra and graph summarization can be employed.

1) Graph Spectra: In order to reduce the size of a big graph, we can map the adjacency matrix of a graph into its spectrum domain, and then a threshold can be decided according to the requirement of given applications to filter out the small elements. For big data representation using graph spectra, we can obtain a smaller and simpler graph from the filtered graph spectrum through inverse transformation. A comprehensive introduction to graph spectra with regard to complex networks can be found in [58].

As an important data source of big data applications, social networks have received much attention in recent years. To understand and utilize the information in social networks, researchers have developed various measures to indicate the structures and characteristics of networks from different perspectives [59]. To date, various properties including its size, density (a measure of the relative number of connections), power-law degree distributions, average distance, small-world phenomenon, clustering coefficient (the tendency of a network to aggregate in subgroups), community structures have been discovered. Among these features, Ying *et al*. [60] firstly investigated the issues of randomness versus non-randomness in social networks. The authors theoretically analyzed graph randomness and presented a framework which provided a series of non-randomness measures at the levels of edge, node, subgraph, and the overall graph. By utilizing the spectra of the adjacency matrix of a network, they obtained the graph non-randomness. They also studied whether other graph spectra [61], [62] (such as Laplacian and Normal spectra) could be used to derive non-randomness for social networks.

Epidemics are critical phenomena of big data, not only from a biological viewpoint, such as infectious diseases, but also from a technological viewpoint, such as malware propagation. Therefore, it is imperative to develop accurate and effective models for epidemics. In most existing individual-based epidemic models [63], the interaction is driven by a single graph. However, studying epidemics in communication networks and cyber-physical systems requires a more elaborate description of interactions. Motivated by this challenge, Sahneh *et al*. [64] provided a detailed description of the stochastic process at the agent level where the agents interact through different layers, each represented by a graph. The structure of the proposed model is characterized by the elements of the adjacency matrices of the network layer, and the Laplacian matrices (Laplacian spectra) of the transition rate graphs.

Predicting the popularity of online content has been a subject of great interest due to its evergrowing importance in big data applications, such as network content caching and advertising. Previous research on online media popularity prediction concluded that the rise in popularity of online videos maintains a conventional logarithmic distribution. However, recent studies have shown that a significant portion of online videos exhibit bursty or sudden rise in popularity [65], [66], which cannot be accounted by video domain features alone. Roy *et al.* [67] proposed a novel transfer learning framework that utilizes knowledge from social streams (e.g., Twitter) to grasp sudden popularity bursts in online content. By employing spectral analysis, the authors represented both the social and the video feature information as a combined feature representation. Extensive experiments on real world data comprising of 10.2 million tweets and 3.5 million YouTube videos showed that the proposed transfer learning algorithm is quite effective.

2) Graph Summarization: The database community has accumulated a technique named graph summarization [68], which can also be used for big data representation. Existing graph summarization methods [69]–[71] are mostly statistical (i.e., studying statistics such as degree distributions, hop-plots and clustering coefficients). From users' perspective, the summarization method should allow them to freely choose the attributes and relationships that are of interest, and then make use of these features to produce small and informative summaries. Motivated by this challenge, Tian *et al.* [68] introduced two database-style operations to summarize graphs. Like the OLAP (On-Line Analytical Processing) style aggregation methods that allow users to drill-down or roll-up to control the resolution of summarization, the proposed method provides an analogous functionality for large graph data sets.

In real-life settings, large graphs are quite common, such as Web graphs, social networks, and computer communication graphs. In order to find patterns in a large graph, it is desirable to mine, compute, and visualize it. However, it is difficult to deal with graphs with hundreds of thousands of nodes and millions of edges. For example, the excessive processing requirements are prohibitive; drawing hundred-thousand nodes results in cluttered images that are hardly to comprehend [72]. To address these problems, Rodrigues *et al*. [72] proposed an innovative framework (GMine) suited for any kind of tree-like graph. As a proof of concept, the visual environment of GMine is instantiated as a system in which large graphs can be investigated globally and locally.

As an application case of big data network, Shi *et al.* [73] investigated citation networks. The authors tried to address the Influence Graph Summarization (IGS) problem: How to make sense of an individual's influence in the context of the citation network. Particularly, how to summarize the underlying citation graph to represent this influence. A matrix decomposition based algorithm was proposed to solve the IGS problem. Through this summarization scheme, the impact of a highly influential paper can be easily highlighted. Comprehensive experiments using real-world citation networks demonstrated that the proposed method significantly outperforms the previous ones in optimizing both the quantitative IGS objective and the quality of the visual summarizations.

3) Graph Compression: How can we compress graphs efficiently? How can we find communities in graphs? The two questions are closely related: if we could find a good method to identify communities [75], then we are able to compress the graph well since the nodes in the same community possess redundancies (e.g., similar neighborhood) which help us shrink the size of the data. For community detection, the traditional research focus was on finding homogeneous regions in graphs, i.e., the "caveman communities", so that cross edges between different regions are minimize. However, recent results based on real world networks showed that real world graphs generally do not have good cuts [76].

For graph compression, it has been widely applied in various areas, such as Web graphs [77] and social networks [74]. However, literatures almost all neglect the fact that real world graphs are much more complicated and inter-connected than caveman graphs. It is well known that most real world graphs follow power-law distributions with few 'hub' nodes having very high degrees and majority of the nodes having low degrees [78]. It is also known that a significant proportion of the hub nodes effectively combines many caves into a huge cave [79], which breaks the assumption of the caveman-like community structure. Lim *et al.* [75] proposed to exploit hubs and the neighbors of the hubs to define a community for graph compression. The proposed method is based on the observation that graphs are easily disconnected by hubs, or high degree nodes: removing hubs from a graph creates many small disconnected components, and the remaining giant connected component is substantially smaller than the original graph. The proposed approach is much more suitable for real-world, power-law graphs. Extensive experiments on real-world graphs confirmed that it gives good compression results.

We summarize and compare the available techniques of graph representation for big data in Table I.

B. Dynamic Representation of Big Data

So far, we have reviewed recent studies on the static representation of big data. However, plenty of big data networks are significantly featured by dynamics, such as the Internet. In other words, network topology and properties are time varying. For example, nodes come and go, connections between nodes vary from time to time. It is obvious that new representation tools to represent these dynamic networks are highly expected. In the following of this subsection, we review the existing work on dynamic representation of big data.

Researchers refer to dynamic networks as networks with a node set V and an edge set E, as well as sets of node attributes V_A and edge weights E_W being subject to change over time [80]. During the course of the exploration of dynamic networks, visual representations are often switched as the focus of analysis shifts between the temporal and the structural aspects of the data. To support such switching in a seamless and intuitive manner, Hadlak *et al*. [80] proposed a novel scheme that tightly integrates existing visualization techniques for dynamic networks. For large dynamic networks, the number of data items is reduced beforehand by either reducing the size of the network or reducing the number of time steps. The level of reduction of each aspect is chosen so as to reflect the focus of the visual analysis: if the focus of analysis lies on the network aspect, users may not want to reduce the network structure, e.g., by clustering, but instead rather cut down on the number of time steps. Likewise, if the analysis is centered around the temporal aspect of the dynamic network, the number of time steps should be remained as many as possible, while aggregating the network instead.

The temporal evolution of a graph introduces another interesting aspect to graph drawing, but it also brings further visualization challenges concerning the additional time dimension. Burch *et al.* [81] introduced a scalable single-image visualization technique for exploring dynamic and hierarchically organized graphs. They mapped each time step of the graph to a narrow rectangular area on screen. The vertices were arranged along the vertical borders of these areas, according to the order implied by the hierarchy. Directed graph edges connected the vertices by straight links starting on the left hand side of the area and ending at the right hand side. In large graphs, this strategy leads to a plethora of overlapping edges. The authors tackled this problem by applying parallel edge splattingała technique that transforms the edges to a density scalar field by splatting those edges to the screen. Visualizing this field by color mapping allows us to recognize the trajectories of edges even in quite cluttered areas. The term "parallel" indicates the layout of graph vertices on parallel lines. Moreover, the proposed visualization tool provides several interactive features to manipulate and navigate dynamic graph data.

TABLE I
APPROACHES OF GRAPH REPRESENTATION FOR BIG DATA

Approaches	Addressed Issue	Proposed solution	Pros	Cons
SBF [60]	Capture randomness and non-randomness in social networks	Spectra analysis of the adjacency	- Outperforms Laplacian and normal spectra	- High computational complexity issues for large social networks
GEMF [64]	Model dynamic behavior of epidemic spreading processes	Generalized Epidemic Mean-Field Model based on Laplacian matrixes of the transition rate graphs	- Simple to apply in many applications	- Accuracy depends on the range of the epidemic parameters - Sensitive to the initial states
TLF [67]	Social video popularity prediction	Transfer learning by employing spectra analysis	- Scalability - High performance - Predict bursty videos	- Media application specific
GS [69], [70]	Summarization of large graph	Graph summarization based on statistics	- Efficiency	- Summarization contain limited information, hard to interpret and manipulate
SNAP [68]	Summarization of large graph	Database-style graph aggregation operations	- Summarization easy to control	- Memory bottleneck
CEPS [72]	Summarization of large graph	Center-piece subgraphs	- Scalability - Visual summarization	- User interaction
VEGAS [73]	Analyze citation networks	Matrix decomposition	- High performance	- Application specific - Scalability
CSN [74]	Compress social networks	Compression on caveman graphs	- Study the extent to which a large network can be compressed in detail	- Assumption of caveman like community structure
SlashBurn [75]	Graph compression beyond caveman community	Considering the power law characteristics of real graph	- More practical - Good compression results - Fast running time	- Require square blocks stored in a distributed manner

Maps offer a familiar way to present geographic data (continents, countries), and additional information (topography, geology) can be displayed with the help of contours and heat-map overlays. Mashima *et al.* [82] considered visualizing large-scale dynamic relational data by taking advantage of the geographic map metaphor. They described a map-based visualization system which used animation to convey dynamics in large data sets. The proposed system aimed to preserve the viewer's mental map while also offering readable views at all times. It was fully functional and had been used to visualize user traffic on the Internet radio station last.fm, as well as TV-viewing patterns from an IPTV service [82].

In current stage, the state-of-the-art works on dynamic representation of big data mainly focus on visualization technique. With big data becoming more popular, it is obvious that researchers should try other promising tools for this topic. Besides the visualization representation for dynamic big data networks, the mining community also has been working on this topic, named probabilistic graphical model [83]. However, based on our understanding it is far from practical to be used in the networking field. We will further discuss the relevant issues in Section V.

C. Matrix Representation of Big Data

1) Singular Value Decomposition: The techniques for matrix factorization have become popular in recent years for data representation. In many problems of information retrieval, computer vision, and pattern recognition, the input data matrix is of very high dimension, researchers then hope to find two or more lower dimensional matrixes whose product provides a good approximation to the original one [84]. Singular value decomposition is one of the most frequently used matrix factorization techniques. A singular value decomposition of an $M \times N$ matrix $\mathbf{X}$ has the following form [85].

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T,$$

where $\mathbf{U}$ is an $M \times M$ orthogonal matrix, $\mathbf{V}$ is an $N \times N$ orthogonal matrix, and Σ is an $M \times N$ diagonal matrix with $\Sigma_{ij} = 0$ if $i \neq j$, and $\Sigma_{ii} \geq 0$. The quantities Σ_{ii} are called the *singular values* of $\mathbf{X}$, and the columns of $\mathbf{U}$ and $\mathbf{V}$ are called left and right *singular vectors*, respectively. By removing those singular vectors corresponding to sufficiently small singular values, we obtain a low-rank approximation to the original matrix. This approximation is optimal in terms of reconstruction error, and thus optimal for data representation when euclidean structure is concerned. For this reason, SVD has been applied in various real-world applications, such as face recognition [86] and document representation [87].

2) Nonnegative Matrix Factorization: Previous studies have shown that there is psychological and physiological evidence for parts-based representation in the human brain [88], [89]. The Nonnegative Matrix Factorization (NMF) algorithm was proposed to learn the parts of objects like human faces and text documents [90], [91]. NMF aims to find two nonnegative matrices whose product provides a good approximation to the original matrix. Concretely, given an $M \times N$ matrix $\mathbf{X}$, NMF aims to find two nonnegative matrixes $\mathbf{U}$ and $\mathbf{V}$ whose product can well approximate the original matrix $\mathbf{X}$:

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}^T,$$

where $\mathbf{U}$ is an $M \times K$ matrix, and $\mathbf{V}$ is an $N \times K$ matrix.

All the three matrixes $\mathbf{X}$, $\mathbf{U}$, $\mathbf{V}$ have no negative elements. This non-negativity is inherent to the data being considered in many big data applications, such as location recommendation [92], traffic estimation [93]. The nonnegative constraints lead to a parts-based representation because they allow only additive, no subtractive or combinations. NMF has been shown to be superior to SVD in various areas, such as face recognition [94] and document clustering [95]. Moreover, SVD is more computationally expensive and harder to parallelize compared to NMF.

From the geometric perspective, the data is usually sampled from a low-dimensional manifold embedded in a high-dimensional ambient space. Researchers then hope to find a compact representation, which uncovers the hidden semantics and simultaneously represents the intrinsic geometric structure. To address this problem, Cai *et al.* [84] proposed graph regularized NMF. An affinity graph is constructed to encode the geometrical information, and the authors seek a matrix factorization, which represents the graph structure. Extensive empirical study on real-world problems showed that the proposed algorithm outperforms the state-of-the-art algorithms.

NMF is essentially an unsupervised method and cannot make use of label information. Liu *et al.* [96] proposed to incorporate the label information as additional constraints with NMF. Specifically, they investigate how explicitly combining label information to improve the discriminating power of the resulting matrix decomposition. For big data applications (e.g., signal processing), dimensionality reduction has been widely studied in the communities. One of the major drawbacks of the existing popular approaches is the lack of physical meaning in the reduced dimension space. In view of this challenge, Esser *et al.* [97] presented a framework for dimensionality reduction, based on matrix factorization and sparsity theory. In order to guarantee physical fidelity, they used the data itself (or small variations from it) for the low-dimensional representation.

A comprehensive review on recent theoretical research of NMF can be found in [102]. Specifically, the paper focuses on the principles, basic models, properties, and algorithms of NMF along with its various modifications, extensions, and generalization. It constructs an integrated, state-of-the-art framework for NMF concept, open issues, and relevant application areas.

3) Tensor Techniques: A tensor is a multidimensional array. More specifically, an Nth-order tensor is an element of the tensor product of N vector spaces, each of which has its own coordinate system [103]. Specifically, a second-order tensor is a matrix.

Variety and veracity are two distinct characteristics of large-scale and heterogeneous data. It has been a great challenge to efficiently represent and process big data with a unified scheme. Kuang *et al.* [98] proposed a unified tensor model to represent the unstructured, semi-structured, and structured data. Concretely, the paper aimed to address the following two issues: (1) how to represent various types of data with a simple tensor model; (2) how to extract the core data sets which are smaller but still contain valuable information, especially for streaming data. A case study illustrates that approximate data reconstructed from a core set containing 18% elements can guarantee 93% accuracy in general.

As an application of big data, video retrieval and indexing research aims to efficiently and effectively manage very large video databases. With the explosive growth of digital video data, shot-based retrieval serves as the basis for video retrieval. How to meaningfully represent shots is a crucial issue. For this purpose, Gao *et al.* [99] proposed to represent video data by developing an optical flow tensor. Concretely, tensors are utilized to provide a new representation of motion features based on optical flow fields.

Similarly, tensor techniques also have been applied in other big data applications, i.e., computer vision. Graph-embedding along with its linearization and kernelization provides a general framework that unifies most traditional dimensionality reduction algorithms. From this framework, Li *et al.* [100] proposed a new manifold learning technique called discriminant locally linear embedding. To deal with the out-of-sample problem in visual recognition with vector input, a multilinear version is proposed with high-order tensor input. Based on the image object representation, the algorithms for image processing utilizing higher order tensor decomposition can be roughly classified into two categories, *image-as-vector* [104], [105] and *image-as-matrix* [106], [107]. Yan *et al.* [101] presented a novel approach to solve the supervised dimensionality reduction problem by encoding an image object as a general tensor of second or higher order. Firstly, they proposed a discriminant tensor criterion, whereby multiple interrelated lower-dimensional discriminative subspaces were derived for feature selection. Then a novel approach called k-mode cluster-based discriminant analysis was presented to iteratively learn these subspaces by unfolding the tensor along different tensor dimensions. Extensive experiments encoding face images as second or third order tensors demonstrated that the proposed algorithm outperformed traditional subspace learning algorithm.

4) Compressed Sensing: The Compressed Sensing (CS) theory can significantly reduce the number of sampling points that directly corresponds to the volume of data collected, which means that part of the redundant data is never acquired. It makes it possible to create standalone and net-centric applications with fewer resources required. Donoho [108] detailed the procedure of compressed sensing. More specifically, suppose x is an unknown vector in $\mathbf{R^m}$ (e.g., a digital image or signal). We plan to measure n general linear functionals of x and then reconstruct x. If x is known to compressible by transform coding with a known transform, and we reconstruct x via the nonlinear procedure defined, the number of measurements n can be dramatically smaller than the size m.

In order to add more flexibility to adapt the representation to the data, researchers have turned to linear decomposition of data based on dictionary learning. The classical approach to learn dictionary is data driven, and the relevant communities have shown that better results can be obtained if the dictionary is task-driven [109]. As a first step towards this challenge, Duarte-Carvajalino and Sapiro [110] proposed to learn dictionaries for signal data using CS. They introduced

TABLE II
APPROACHES OF MATRIX REPRESENTATION FOR BIG DATA

Approaches	Addressed Issue	Proposed solution	Pros	Cons
GNMF [84]	Data representation with the intrinsic geometric structure	Graph Regularized NMF	- Better performance than standard NMF - Better representation in the sense of semantic structure	- Unclear how to select model parameter λ - Can not guarantee data points from the same class will be mapped together
CNMF [96]	Image representation with label information	Incorporate label information as hard constraints	- Parameter free - Guarantee data points from the same class will be mapped together	- Lack of physical meaning in the reduced dimension
NMF-ST [97]	Lack of physical meaning in dimensionality reduction	Combine NMF and sparsity theory	- Guarantee physical fidelity	- Application specific
IHOSVD [98]	Represent big data with a unified scheme	Various types of data represented as subtensors and merged to a unified tensor	- A unified model integrating structured, semi-structured, and structured data	- Approximation error - Scalability
OFT [99]	Video retrieval in very large video database	Represent video data by an optical flow tensor and incorporate HMM	- Speed up video retrieval	- Application specific - scalability
DLLE [100]	Dimensionality reduction for images	Higher order tensor decomposition with label informaton and manifold structure	- High performance	- Unclear how to select optimal parameters - Application specific
DATER [101]	Dimensionality reduction for images	Encoding image object as higher order tensor	- Low computation cost	- Application specific - Scalability

a framework for the joint design and optimization from a set of training images. Extensive experiments on image data sets confirmed that the joint design of CS and dictionary learning outperforms the methods which only rely on dictionary learning or CS.

The application procedure of CS theory is composed of two parts: data acquisition and reconstruct from the sensory data. In Wireless Sensor Networks (WSNs) and Internet of Things (IoT), the two parts are perfectly integrated as a whole process. WSNs and IoT have been widely adopted to collect sensory data and reconstruct the environment in the cyber space. Thus, a low-cost data acquisition system is expected to effectively collect and process the data in WSNs and IoT. Generally, a certain number of samples of data is able to enable capturing all required information in transformation process without loss of information [111]. Inspired by this property, the authors thoroughly investigated how CS can provide new insights into data sampling and acquisition [112]. A compressed sensing-based framework was proposed for IoT, where the end nodes measure, transmit, and store the sampled data. In order to achieve accurate data reconstruction and lower energy efficiency, an efficient cluster-sparse reconstruction algorithm was then established.

Another property of WSNs is that data loss is common and has its special patterns due to noise, collision, unreliable link, and unexpected damage [113]. As a result, it is important to reconstruct the environment by sensory data, which is a fundamental operation for understanding the big data in depth. In [114], however, the authors pointed out that CS-based methods require the data set to have inherent structures. Moreover, CS theory performs well when the missing values follow the Gaussian or pure random distribution, which is not satisfied by the data loss pattern of WSNs. Thus, existing CS-based interpolation methods cannot be directly applied for accurate environment reconstruction. To address this problem, Kong *et al*. [114] proposed a novel approach based on CS to reconstruct the massive missing data. By analyzing the real sensory data from Intel Indoor, GreenOrbs, and Ocean Sense projects, the authors found that all the data exhibit the features of low-rank structure, spatial similarity, temporal stability and multi-attribute correlation. Based on these observations, they developed an environmental space time improved compressive sensing algorithm with a multi-attribute assistant component for data reconstruction. Extensive experiments on real sensory data sets showed the proposed method significantly outperforms existing solutions in terms of reconstruction accuracy.

In Table II, we summarize and compare the existing works on matrix representation for big data applications. As NMF generally outperforms SVD, we neglect the works on SVD for clarity.

IV. NETWORKING TECHNOLOGIES SERVING BIG DATA APPLICATIONS

As networking for big data is still at an early stage, we mainly survey literatures on three existing techniques in the scenario of big data applications: extended queueing theory for big data applications, big data scheduling, and systematic modeling for dig data.

A. *Extending Queueing Theory for Big Data Applications*

Queueing theory are proposed to tackle the following problems. Customers request the use of a particular type of services. If a server is available, the arriving customer will seize and hold it for some length of time, after which the server will be made immediately available to other incoming or waiting customers. If an incoming customer finds no available server, it then takes some specified action such as waiting or going away. Such models often can be defined in terms of three characteristics: the input process, the service mechanism, and the queue discipline. More specifically, we employ the widely used notation $a/b/c$ to describe queueing models [115]. In this notation, a specifies the arrival process, b specifies the service time, and c is the number of servers. For example, $M/M/1$ denotes a queueing model with Poisson input, exponential service time distribution, and 1 server. Queueing theory has been widely used for performance evaluation [116]–[118].

Performance modeling for big data systems introduces many new challenges. As information retrieval is always an important part for big data applications, for example, we usually use the ideal $M/M/$ model to approximate the real situation in practice when data scale is not that big. However, we cannot employ this approximation any more in the big data case. In terms of service rate, if the searched object is homogeneous, such as records of people, then it is a geometric distribution following a sequential search process [119]. Specifically, let p_i denote the probability of searching succeeds at a particular item (data) D_i. As the searched item is homogeneous, we can assume that all the item share the same probability of searching, i.e., $p_1 = p_2 = \cdots = p_i = \cdots = p$. Let random variable Y denote the number of trials for the first match, then the probability that we obtain item D_i follows a geometric distribution, which is expressed as

$$Pr[Y = i] = \begin{cases} (1-p)^{i-1}p & 0 < i < N \\ (1-p)^{N-1} & i = N. \end{cases}$$

On the other hand, if the searched item is a heterogeneous, such as Web files, then the distribution should be power law [120]. To date, these problems fall in the category of the $G/G/$ model. However, we do not have closed form solutions for the $G/G/$ model due to the fact that there is no mathematical expression of a specific G.

Though the general model $G/G/$ is intractable, approximations are generally available. In big data applications, tasks or jobs are usually carried out by multiple servers in a parallel or distributed way. For example, in order to identify the trace of a malicious car in a city, we need to check many traffic camera records. From a global view, it is generally required to optimize some performance metrics for this distributed system, such as maximum throughout, and minimal delay. Taking these global optimization requirements into consideration, the traditional queueing theory can be extended for big data applications performance modeling. In the following, we review the existing works on performance modeling for big data applications using extended queueing theory.

Wang *et al.* [121] proposed a novel queueing architecture for task scheduling in MapReduce, which integrated data locality in the queueing model, i.e., scheduling the map tasks in local or remote machines as shown in Figure 1. In this queueing architecture, there is one local queue for each machine, storing local tasks associated with the machine, and a common

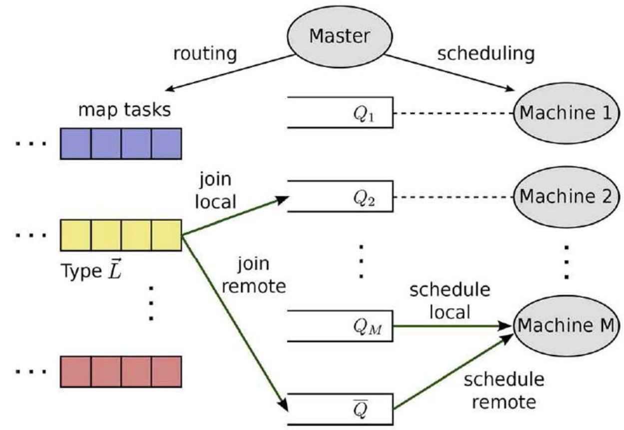


Fig. 1. Queue architecture: Q_1, Q_2, ..., Q_M are local queues associated with machines $1, 2, \ldots, M$, respectively, and $\overline{Q}$ is the common remote queue. The scheduling algorithm has two steps: routing and scheduling [121].

queue for all machines. Based on this new queueing architecture, they proposed a two-stage scheduling algorithm under which a newly arrived task is routed to one of the three local queues associated with the three local machines or the common queue using the Join the Shortest Queue (JSQ) policy; when a machine is available, it selects a task from the associated local queue or the common queue using the MaxWeight policy [121]. The MapReduce framework has been widely used to process large-scale data sets on computing clusters. Scheduling map tasks with data locality consideration is crucial to the performance of MapReduce. However, this issue has not been well studied so far. The proposed extended queueing model is proved to strike the right balance between data locality and load balancing to simultaneously maximize throughput and minimize delay.

Many applications in wireless sensor networks are data oriented [122]. Data collection in sensor networks typically relies on the wireless communications between sensor nodes and the sink, which may excessively consume the limited energy supply of sensor nodes due to the superlinear path loss exponents [123]. Mobility-assisted data collection in sensor networks creates a new dimension to reduce and balance the energy consumption for sensor nodes. However, it also introduces extra latency in the data collection process due to the limited mobility of the mobile elements. He *et al.* [124] then proposed an extended queueing model to schedule the movement of mobile elements throughout the field. Specifically, the authors investigated the on-demand scenario where data collection requests progressively arrive at the mobile element. The data collection process is modeled as an $M/G/1/c - NJN$ queuing system with an intuitive service discipline of nearest-job-next (NJN). The simulation results confirmed that the proposed queuing model outperforms other scheduling algorithm with the first-come-first-serve (FCFS) discipline.

As a fundamental networking technology, clouds has been widely used for big data applications. In order to capture the data retrieving delay, researchers mainly adopt queueing theory to establish mathematical models. Chen *et al.* [125]

pointed out that the delay is quite random, and exhibits weak correlations between different read/write requests. Thus, it is possible to reduce the delay by leveraging multiple parallel threads. In the proposed model, there is a multiple server (corresponding to multiple threads) queue with arrivals of data retrieving requests. Due to the redundancy created by the threads, a server can terminate its operation depending on when other servers complete their services. To analyze this extended queuing model, the authors presented approximations based on real traces from Amazon S3. The results showed that the downloading time for any individual thread can be approximated as an *i.i.d.* exponentially distributed random variable.

B. Big Data Scheduling

Many big data applications perform big and complex tasks. From the network perspective, these tasks typically compromise multiple flows, which traverse different parts of the network at potentially different times. This task level scheduling is called coflow scheduling. In mathematical expression, there are n distributed flows within a task. We expect the n ($n > 1$) distributed flows ideally completed at the same time point t, and the value of t is as small as possible.

In general, big data scheduling belongs to a mathematical branch called *fork-join queueing systems*. A recent comprehensive survey on this topic can be found in [126]. Another mathematical tool, *network calculus*, may be a promising tool for big data scheduling. Interested readers can refer to the classic textbooks [127], [128] and the latest work at top conferences, such as [129].

In order to achieve performance (e.g., cluster utilization, job completion times) optimization, traditional scheduling algorithms execute at per-flow level [130], [135]. In [131], however, the authors argued that to maximize job performance, we need to optimize at the level of transfers, instead of individual flows. For example, in frameworks like MapReduce and Dryad, a stage cannot complete (or sometimes even cannot start) before it receives all the data from the previous stage. Thus, the job running time depends on the time it takes to complete the entire transfer, rather than the duration of individual flows comprising it. The authors defined a transfer as a set of all flows transporting data between two stages of a job. Then they proposed an control architecture named Orchestra, in which data movement within each transfer is coordinated by a Transfer Controller (TC). The TC continuously monitors the transfer and updates the set of sources associated with each destination.

Dogar *et al.* [28] further confirmed that it is necessary to optimize performance metrics at task level. They designed and implemented Baraat, a decentralized task-aware scheduling system, which groups flows of a task and schedules them together. Baraat schedules tasks in a FIFO order but avoids head-of-line blocking by dynamically changing the level of multiplexing in the network. Extensive experiments showed that Baraat outperforms Orchestra with regard to the average and tail completion time for a wide range of workloads of data center applications.

Chowdhury *et al.* [29] addressed inter-coflow scheduling for two different objectives: decreasing communication time of data-intensive jobs and guaranteeing predictable communication time. They introduced the concurrent open shop scheduling with coupled resources problem, analyzed its complexity, and proposed effective heuristics to optimize either objective. The proposed system Varys enables data-intensive frameworks to use coflows and the proposed algorithms while maintaining high network utilization and guaranteeing starvation freedom. Compared to the per-flow scheduling mechanism, Varys completed up to 3.16 times faster on average and up to 2 times more coflows meet their deadlines in the communication stage.

Minimizing the average coflow completion time (CCT) of applications becomes a critical objective for flow scheduling in current data centers. Luo *et al.* [132] argue that the state-of-the-art centralized method, Varys, achieves a good average CCT; but it has the scalability problem. While the decentralized method, Baraat, suffers from the head-of-line blocking problem. To solve these problems, they proposed a preemptive, decentralized, coflow-aware scheduling scheme. The proposed method pursues coflow-level remaining-time-first (MRTF) principle by leveraging a simple negotiation mechanism between each coflow's data senders and receivers. As the MRTF principle is inherently preemptive and proven to be a near-optimal guideline to minimize average CCT, the method avoids the head-of-line blocking problem and obtains good performances. Simulation results demonstrated that their method achieves a performance close to Varys (gap $< 15\%$) and outperforms Baraat significantly (about $1.4 - 4$ times).

Zhao *et al.* [133] pointed out that routing and scheduling must be jointly considered at the level of a coflow rather than individual flows to optimize application performance. However, prior solutions only considered scheduling. To this end, the authors proposed Papier as a novel coflow-aware network optimization system for data center networks. To improve average CCT, Rapier seamlessly combined routing and scheduling together by formulating it as a joint optimization model. Their evaluation results showed that Rapier can reduce average CCT up to 79.30% and 60.43%, compared to the scheduling-only and routing only schemes, respectively. Moreover, Rapier can be readily implementable with existing commodity switches.

Inter-coflow scheduling improves application-level communication performance in data-parallel clusters. All these methods require a prior knowledge of coflow information (e.g., the number of flows, flow sizes of a coflow). However, the coflow characteristics are actually unknown in advance in most cases. Moreover, the existing methods ignore cluster dynamics like pipelining, task failures, and speculative executions. Consequently, they are hardly to be implemented in practice [134]. Chowdhury and Stoica [134] presented Aalo, a system that strikes a balance and efficiently schedules coflows without prior knowledge. Aalo employs Discretized Coflow-aware Least-Attained Service (D-CLAS) to separate coflows into a small number of priority queues based on how much they have already sent across the cluster. By performing prioritization across queues and scheduling coflows in the

TABLE III
SUMMARY OF LITERATURES ON BIG DATA SCHEDULING

Approaches	Coflow-aware	Performance metrics	Routing	Priority knowledge of coflow
Pfabric [130]	No	Single flow completion time	No	–
Orchestra [131]	Yes	Coflow completion time	No	Yes
Baraat [28]	Yes	Average/Tail coflow completion time	No	No
Varys [29]	Yes	Decrease communication time of data-intensive jobs guarantee predictable communication time	No	Yes
D-CAS [132]	Yes	Average coflow completion time	No	Yes
Rapier [133]	Yes	Routing Average coflow completion time	Yes	Yes
Aalo [134]	Yes	Average coflow completion time	No	No

FIFO order within each queue, Aalo's scheduler reduces coflow completion times while guaranteeing starvation freedom. EC2 deployments and trace-driven simulations showed that communication stages complete 1.93 times faster on average, and 3.59 times faster at the 95th percentile using Aalo in comparison to per-flow mechanisms [134]. Aalo's performance is comparable to that of solutions using prior knowledge, and it outperforms them in presence of cluster dynamics.

In Table III, we summarize the existing methods on coflow scheduling. We mainly compare them from three aspects, coflow-aware or not, information-aware or not, routing-aware or not. For "information-aware", we refer to "whether they rely on priori knowledge of coflow".

C. Systematic Modeling for Big Data Applications

In this part, we aim to review the modeling techniques for big data at a system level. By "system level" we mean that the big data systems are treated as black boxes, and we conduct performance analysis based on system functions like the input and output function.

System science has been developed for many years, however, this cherished method has not been employed effectively in networking community. In general, there is an input $x(t)$ of a system, which is usually treated as a black box with a system function $h(t)$. Then the output of the system is $y(t) = x(t)*h(t)$ (Here $*$ denotes the convolution operation). In order to obtain a closed form solution of $y(t)$, a common method is transforming $x(t)$ and $h(t)$ into a feature domain via different domain mapping techniques, such as Laplace transform, to obtain $Y(s)$, and then obtain $y(t)$ from $Y(s)$ through an inverse transform. In the big data case, we will often face power law distribution, which is pervasive in nature and man-made systems. For example, a statistical framework demonstrated that empirical data generally follows power law distribution [136]. People also found that the size distribution usually follows the power law, such as the population in cities in a country or personal income in a nation [137]. In terms of the Internet, researchers have discovered many power law phenomenon, such as the size distribution of Web files [120]. Recent progresses reported in [138] further demonstrated that the size of networks follows the power law.

Currently, the Pareto distribution has been commonly used as a system function for big data applications. Holm [139] brought important insight into this aspect through analysis of $5,602,097$ malware alarms corresponding to $203,025$ intrusions, which have occurred across $261,757$ computer systems of a large international enterprise between October 2009 and August 2012. The results showed that the Pareto distribution is a significantly better fit in terms of modeling the time of first intrusion. For software systems, the bugs in them are also Pareto distributed [140]. From the aspect of system modeling, a Laplace transform of the Pareto distribution is highly expected if it is adopted as the input system function. However, Fischer *et al.* [141] concluded that the Laplace transform may not exist in 2001. Approximation methods were proposed later via hyperexponential distributions fitting technique combined with Laplace transform [142]. Rodrfguez-Dagnino [143] obtained the Pareto Laplace transform with many parameters, which is difficult to be applied in practice.

The recent works [120], [144] have proved that the Zipf distribution and the Pareto distribution are the same, and they can be transformed to each other. However, the Zipf distribution possesses a simpler math expression. Consequently, we have witnessed a growth of applications of the Zipf distribution. In the study of scheduling algorithms on large-scale parallel and distributed systems, workload modeling and performance evaluation play quite crucial roles. By analyzing the common presence of numerous workload features in real data, the authors pointed out that the current workload models are generally unrealistic as they do not capture the features like periodicity and temporal burstiness of job arrivals, bag-of-tasks behavior [145]. Then they proposed a systematic approach to model parallel workloads by capturing all relevant features, where the bag-of-tasks size follow a Zipf distribution. Wang *et al.* [146] tried to theoretically understanding the push-based content delivery in a converged broadcasting and cellular network to relieve the burden caused by the fast growing wireless data traffic. In order to analyze the scheme in which the most popular contents are pushed through broadcasting to alleviate the cellular data bottleneck, the popularity of the multimedia contents are modeled as Zipf distributed.

The existing works about systematic modeling using convolution operation covers various aspects. Yu *et al.* [48] investigated how malware propagate in networks from a global perspective. Specifically, the authors found that the probability distribution function (pdf) of number of infected hosts equals the convolution of the pdf of number of networks that have been compromised and the pdf of the size of a compromised network. Misic and Misic [147] focused on the activity patterns of primary users in cognitive networks. They established a system model where the pdf of the total cycle time on a channel is the convolution of the pdfs of the corresponding active and idle times of the primary user using this channel. Based on this mathematical model, they got a thorough analysis of the pdf of spectral hole duration in cognitive networks. These results can be used to aid in performance analysis of MAC-level algorithms and protocols for cognitive networks. Tang *et al.* [148] presented a systematic model with convolution operation to characterize the channel impulse response in underwater wireless optical communication (UWOC) links. The receive signal is the convolution of the transmit signal and the impulse response of the links. It is plausible and convenient to utilize this impulse response model for performance analysis and system design of UWOC systems.

Despite all these works, the systematic modeling of big data applications has attracted little attention so far. An essential issue is that we are not sure whether we can derive closed-form expression from the convolution operation or not. With big data becoming more popular, it is significantly necessary to verify whether the big data can be processed within a tolerable "delay" by networking techniques.

V. Research Challenges and Opportunities

Big data poses unprecedented challenges to networking technologies. Though researchers have proposed potential solutions for some of the discussed issues, there are many other open issues still not being sufficiently studied, and need to be further explored. In this section, we review some of the most important open research issues in terms of networking for big data.

A. Networking for Big Graph Mining

Under the background of big data, big graphs are ubiquitous, i.e., ranging from social networks to the World Wide Web and Internet-of-Things. How to mine such big graphs efficiently? How do we find patterns or anomalies in very large graphs with billions of nodes and edges? Although we have reviewed some techniques to compress the big graph to some extent, it is still a tough task to process the input graphs in the memory or disk of a single machine. Since single machine algorithms are intractable, researchers turn to distributed schemes to handle big graphs. In general, graph processing algorithms are iterative and need to traverse the graph in some way [149]. The existing works on distributed big graph mining can be categorized into two types.

MapReduce/Hadoop based systems include Surfer [150], Gbase [151], and Pegasus [152]. However, these approaches remain inefficient for the graph processing case due to the unnecessary wasted I/O operations, network bandwidth, and processor resources [149]. In order to address these problems, non-MapReduce based platforms have emerged recently, mainly include Pregel [153], GraphLab [154], Trinity [155], and Spark [156]. A comprehensive review of the state-of-the-art big graph processing systems can be found in [149].

The elaborate hard work has solved big graph mining to some extent. However, it is still a challenging task to mine big graphs for big data applications, mainly due to their size and the inherent irregular structure of graphs. Take Pregel as an example, it is an efficient, scalable and fault-tolerant system introduced by Google for processing large graphs. As the entire computation state of Pregel resides in the memory, it may require terabytes of memory for very large graphs, which is generally impossible. Currently, graphs with millions and billion of nodes and edges have become very common, which consequently raises heavy resource (e.g., RAM, disk, and bandwidth) burden on the existing algorithms. A more feasible and practical direction is to achieve a tradeoff between resource consumption and performance. In principle, both the computation cost in terms of resource consumption and the computation performance are affected by the structure of node-edge relations in the graphs. Therefore, it is crucial to develop graph representation mechanism (e.g., graph compression, graph summarization as aforementioned in Section III) to optimize both the size and the structure depending on application types. Besides, the partitioning (caused by the representation mechanism) of the input graph based on the topology may suffice if the topology corresponds to the message traffic. Consequently, the representation mechanism need to capture the dynamics in big graphs if necessary.

B. Heterogenous Network Analysis

For homogeneous network analysis, numerous methods have been proposed, such as PageRank, and community structure detection for social networks. As heterogenous networks generally carry rather richer information and semantic meanings, however, most of these methods for homogeneous networks can not be applied directly to analyze their heterogenous counterparts. Therefore, we need to develop novel and powerful analytical model to capture the rich information hidden in the heterogenous links across nodes. As stated in [56], heterogenous network analysis should following three principles, information propagation across heterogeneous types of nodes and links, searching and mining by exploring network meta structures, user-guided exploration of information networks. As the first work towards this topic, the authors also pointed out the major tasks and techniques for heterogeneous networks mining. Beyond these basic tasks, heterogenous network analysis also face several other challenges.

The first issue is how to construct high-quality heterogenous networks from the related big data applications. By "high quality" we mean the network structure is well-defined. For example, a large number of the nodes and links are relatively clean and unambiguous. Facing the complex, unstructured big data applications, techniques like detection of hidden semantic relationship, data cleaning, and natural language processing should be integrated for high-quality heterogenous network construction.

How to enable online processing of heterogenous networks with real time delay is another possible direction for further research. Given the enormous size and complexity of a network for big data applications, processing on the whole network generally takes quite long time to complete. Due to limitations like resource constraints, tasks of this type are generally executed in an off-line manner. However, sometimes users are only interested in a small portion of the networks relevant to their task. In this case, users prefer the "real time delay" to "optimal results". Then how to map users' application requirements into intelligent query and semantic search could be interesting topics.

C. Dynamic Representation in Networking for Big Data

For many big data applications, graph representations are evolving over time. As aforementioned in Section III, existing works [80]–[82] about graph dynamics representation almost all focus on visualization techniques. As graphs for big data applications are quite complicated with large number of interactions between nodes, visualization of graphs does help researchers understand graphs better. All these methods support depicting the dynamic graph in a single view. However, the output views are a bit counter-intuitive, it requires some effort to understand them. How to effectively represent dynamic graphs in a screen with limited resolution is still an open problem.

Integrating time as another dimension in the representation of dynamic graph can lead to interesting discoveries, which are usually hidden in static representations. Another significant issue is that we should capture the rich semantics in mathematical representations. For example, disease propagation through various channels among people, different kinds of animals, and foods in a healthcare network should be distinct. So far, it is quite difficult to describe these semantics in a dynamic representation. A promising approach is to use tensor techniques to model time evolving graphs by using time as the third dimension, and find correlations between dimensions using tensor decomposition [152].

D. Time Evolving in Networking for Big Data

In many real world domains, the processes that generated the data are time evolving. Namely, the data itself is also dynamic. Consequently, the block structures embedded into the time-varying data should also evolve smoothly over time [157]. It is desirable to select features by incorporating the temporal smooth nature of the data into potential networking theories. Li *et al*. [157] developed an evolutionary co-clustering formulation for identifying co-clusters from time-varying data. As proposed by the authors, sparsity-inducing regularization [158] is a promising approach to identify block structures from the time-varying data matrices. However, how to set the regularization parameters adaptively to capture various levels of evolutionary data is still very challenging [159].

In general, large scale data sets are processed in batch. The time evolving feature of data facilitates the possibility of employing the online stream-processing techniques for big data processing [160]. For instance, Luts [161] proposed two approaches for temporal adaptation for real-time semi-parametric regression of distributed data streams, offering fully-automated regularization for evolving environments. Both of them work by assigning different weights for old and new data. However, how to determine the learning rate (to update the statistics for the data in different time window) and reasonable weights is very tough.

In Section II, we have reviewed recent works on epidemic theory to model data dissemination. In essence, they are almost all differential equation based models. However, researchers have discovered that not being able to approximate the infinitesimal condition is the essential reason causing the inaccuracy of the epidemic models [48]. To address this problem, we can extend the existing differential equation based models by introducing probabilistic elements, or establishing novel models by removing the infinitesimal element, such as a pure probabilistic model. Moreover, we have to realize that human behavior [162] is the fundamental reason behind community formation (and therefore the related big data sets). Therefore, we still need to include human behavior and social psychology as a necessary foundation for this research direction.

E. Scheduling in Networking for Big Data

As an emerging popular research topic, coflow scheduling is quite important in network optimizations for big data applications. A common problem with existing scheduling algorithms is that they almost all require a priori knowledge of coflow information, which is actually impractical [134]. How to apply these methods in a real environment without compromising the promising high performance is a good start point.

Although researchers have considered coflow scheduling scheme without prior knowledge [134], there are also a lot of issues unaddressed. For example, how to find the optimal number of priority queues and corresponding thresholds. Due to cross-flow dependencies, recent works in determining similar thresholds in the context of flows [163] can not be directly extended to coflows. Researchers may turn to online learning to dynamically change these parameters. Though we do have enough information with regard to the number of flows, flow sizes of a coflow, it is generally easy to access the statistic characteristics of coflows, such as coflows size distribution, traffic intensity. Inspired by this observation, integrating such type of information surely supports and improves the algorithm design. Finally, how to achieve coflow scheduling with load balancing taken into consideration remain largely unexplored.

F. Security in Networking for Big Data

The security of big data has drawn great attention in recent years [164]. To the best of our knowledge, however, there are few work on big data security associated with networking. From the network perspective, security issues of big data are bought by the fundamental technology such as the data center networks, cloud computing models, and SDN (Software-Defined Networking). For example, the co-residence due to multi-tenancy may cause data breach, and computation breach [165]. The outsourcing technique exposes threats like data loss and dishonest computation in remote servers. Moreover, big data applications based on SDN networking face single point failure against the logically centralized control plane. Cloud computing and SDN promise to handle massive data and intense computation. The challenge is traditional security mechanism may not suffice due to unbearable computation or communication overhead. For instance, it is impractical to hash the entire data set to verify the integrity of the remotely stored data.

big data presents challenges not properly addressed by the existing security techniques that work with a limited volume of traditional data sources (monitoring system, firewall, IDS/IPS). A promising direction is to establish self-evolving threat ontologies. Chung *et al.* [166] proposed to combine cloud and SDN to facilitate network intrusion detection and countermeasures selection. In the paper, the threat ontologies are constructed in the form of attack graphs by the attack analyzer. In order to be self-evolving, the analyzer continually collects VM profiles and attack histories in the control plane. However, the proposed scheme can not work in decentralized networks. Namely, this method should be improved to apply for big data applications.

Automatic detection of stealthy attacks is another important topic. Stealthy attacks are sophisticated attacks tailored to leverage the worst-case performance of the target system through specific periodic, pulsing, and low-rate traffic patterns [167]. big data analytics make it possible for malicious people to identify potential vulnerabilities for stealthy attacks. To the best of our knowledge, few work has been done on this topic.

G. Privacy in Networking for Big Data

Privacy becomes an increasing concern with the development of big data not only in computing aspect, but also in networking perspective. In general, the data community realized the privacy issue in privacy preserving data publishing (PPDP) more than a decade ago [168], which followed by a new research topic of privacy aware learning [169]. Moreover, the recent survey paper by the world leading researchers pointed out that the appearance of big data is the motivation engine for the development of machine learning techniques, at the same time, machine learning also poses critical threat in terms of privacy in big data age [6].

The privacy threat from networking perspective in the big data age has not been fully realized yet. In the big data age, the data collection and accessing definitely will be performed through networks, which maybe more vulnerable against attacks. We can imagine the possible attacks through the current related research topics. First of all, at the data source end, location privacy protection is a hot topic in wireless or mobile networks, such as location privacy of mobile users [170], and location based service privacy [171]. Secondly, at the data user end, the communication channels may release sufficient information to attacks against the accessed data and the users. For example, Web browsing behavior may be use to identify Web viewers even the users using anonymous communication systems [172].

We fully believe privacy attacks and counter attacks from the networking perspective will be a critical battle ground in the forthcoming big data age, and there are many problems to be explored in the field. We refer the interested readers to the latest extensive survey paper on big data privacy [173].

H. Performance Modeling in Networking for Big Data

It is important to model big data applications from a systematic view, i.e., derive theoretical results for metrics like response time, throughput. In general, the mathematical model can be formulated as an optimization problem, which is mainly characterized by four features, optimization goal with regard to performance metrics, input, system service capability, and the constraints. The optimization goal and the constraints are both application-dependent. In big data applications, there are numerous performance metrics, and optimization of one metric may compromise the others. Moreover, it is generally a tough task for users to explicitly specify a quantitative goal. As a consequence, it is challenging to formulate the system goal and constraints. For the input rate and service capability, we know they fall into the "general" distribution. Therefore, researcher still need to address the problem of deriving closed-form expression for the $G/G/$ model, if queuing theory is applied. Though some approximation [115] have been developed for traditional applications, they are far from practical for big data applications. On the other hand, it is generally difficult to decide whether a certain transform technique can lead to traceable solutions if the convolution operation is adopted. Finally, we have to evaluate the systematic model and theoretical findings through extensive experiments on real-world big data sets.

VI. Summary

Big data has gained an unprecedented attention from academia and industry. While reviewing the literature, we found several important papers surveying specific aspects of big data. Although important research issues have been identified by these survey papers, there is no mentioning about relevant aspects with regard to the networking for big data.

In this paper, we elaborate a comprehensive survey on big data and the fundamental networking technologies. We present our definition of networking for big data as the research complementary to application layer big data based mining, learning, and other applications. We survey the latest research achievement in big data definition, formation, representation, and networking techniques for big data applications, including big data scheduling, system modeling, and so forth. We also present the research challenges and opportunities in networking for big data domain.

Based on our understanding of the surveyed topic, we summarize the major lessons we learnt as follows.

- Networking for big data is an indispensable element for big data applications. Due to the distributed nature and the extraordinary large scale, networking support is an essential component to make big data applications feasible and effective.
- Research on networking for big data has to combine various prospective and different levels of big data, rather than an isolated domain of study. Once again, we define networking for big data as a complementary of big data mining or learning. Therefore, the research range of this field is broad, including job scheduling, data storage, data representation, and so on.

- Research on big data will result in improvement of current techniques and theories, at the same time, it will also motivate us to invent new tools and theories to accommodate the unprecedented problems. The appearance of big data applications has transformed the former trivial or ignored studies to hot topics, such as small probability study in statistics, job synchronization in queueing theory, graph summary in large graph, and so on. The output of all these fields will benefit our design and implementation in terms of networking for big data.

We are standing at the doorstep of big data, and networking for big data is a critical field to accommodate the needs of big data applications. As we studied in this paper, there are many new problems and challenges on our way to pervasive and effective big data applications. Furthermore, we believe many more problems and challenges will show up during the progress of the big data journey from both theoretical and practical perspectives. We hope this paper sheds light on this vivid and exciting research branch, and serves as a solid starting point for interested readers for their further exploration on this almost uncharted land.

References

[1] *IBM What Is Big Data: Bring Big Data to the Enterprise*. Accessed on Mar. 3, 2016. [Online]. Available: http://www-01.ibm.com/software/data/bigdata/

[2] X. Wu and S. Zhang, "Synthesizing high-frequency rules from different data sources," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 2, pp. 353–367, Mar./Apr. 2003.

[3] K. Su, H. Huang, X. Wu, and S. Zhang, "A logical framework for identifying quality knowledge from different data sources," *Decis. Support Syst.*, vol. 42, no. 3, pp. 1673–1683, 2006.

[4] H. A. Abbass, *Computational Red Teaming: Risk Analytics of Big-Data-to-Decisions Intelligent Systems*. Cham, Switzerland: Springer, 2014.

[5] L. Wang, H. Zhong, R. Ranjan, A. Zomaya, and P. Liu, "Estimating the statistical characteristics of remote sensing big data in the wavelet transform domain," *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 3, pp. 324–337, Sep. 2014.

[6] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[7] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.

[8] X. Wu and S. Zhang, "Synthesizing high-frequency rules from different data sources," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 2, pp. 353–367, Mar./Apr. 2003.

[9] K. Su, H. Huang, X. Wu, and S. Zhang, "A logical framework for identifying quality knowledge from different data sources," *Decis. Support Syst.*, vol. 42, no. 3, pp. 1673–1683, 2006.

[10] H. Ke, P. Li, S. Guo, and M. Guo, "On traffic-aware partition and aggregation in MapReduce for big data applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 3, pp. 818–828, Mar. 2016.

[11] 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.

[12] M. Chen, S. Mao, Y. Zhang, and V. C. M. Leung, *Big Data—Related Technologies, Challenges and Future Prospects* (Springer Briefs in Computer Science). Cham, Switzerland: Springer, 2014.

[13] S. Yu, X. Lin, J. Misic, and X. S. Shen, *Networking for Big Data*. Boca Raton, FL, USA: CRC Press, 2015.

[14] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 171–209, 2014.

[15] S. Peng, G. Wang, and D. Xie, "Social influence analysis in social networking big data: Opportunities and challenges," *IEEE Netw.*, to be published.

[16] H. Hu, Y. Wen, T. Chua, and X. Li, "Toward scalable systems for big data analytics: A technology tutorial," *IEEE Access*, vol. 2, pp. 652–687, 2014.

[17] *IEEE Network*. Accessed on Mar. 3, 2016. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6863124

[18] *IEEE Network*. Accessed on Mar. 3, 2016. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7293297

[19] *IEEE Network*. Accessed on Mar. 3, 2016. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7389823

[20] S. Yu and K. Liu, "Special issue on big data from networking perspective," *Big Data Res.*, vol. 3, p. 1–40, Apr. 2016.

[21] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[22] N. McKeown *et al.*, "Openflow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[23] J. Koch, C. L. Staudt, M. Vogel, and H. Meyerhenke, "An empirical comparison of big graph frameworks in the context of network analysis," arXiv:1601.00289, 2016.

[24] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[25] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[26] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, vol. 74, no. 1, p. 47, 2002.

[27] P. Erdös and A. Rényi, "On random graphs I," *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.

[28] F. R. Dogar, T. Karagiannis, H. Ballani, and A. Rowstron, "Decentralized task-aware scheduling for data center networks," in *Proc. ACM Conf. SIGCOMM*, Chicago, IL, USA, 2014, pp. 431–442.

[29] M. Chowdhury, Y. Zhong, and I. Stoica, "Efficient coflow scheduling with varys," in *Proc. ACM Conf. SIGCOMM*, Chicago, IL, USA, 2014, pp. 443–454.

[30] J. Manyika *et al.*, *Big Data: The Next Frontier for Innovation, Competition, and Productivity.* New York, NY, USA: McKinsey & Company, 2011.

[31] C. Stamford, *Gartner's 2013 Hype Cycle for Emerging Technologies Maps Out Evolving Relationship Between Humans and Machines*, Gartner, Stamford, CT, USA, vol. 19, Aug. 2013.

[32] F. X. Diebold, "'Big data' dynamic factor models for macroeconomic measurement and forecasting," in *Advances in Economics and Econometrics: Theory and Applications, Eighth World* Congress (Econometric Society Monographs), M. Dewatripont, L. P. Hansen, and S. J. Turnovsky, Eds., Cambridge, U.K.: Cambridge Univ. Press, 2003, pp. 115–122.

[33] D. Laney, *3D Data Management: Controlling Data Volume, Velocity and Variety*, META Group, Terni, Italy, vol. 6, 2001, p. 70.

[34] H. Barwick. (2008). *The 'Four vs' of Big Data*. [Online]. Available: http://www.computerworld.com.au/article/396198/iiis_four_vs_big_data

[35] *IBM Big Data and Information Management*. Accessed on Mar. 3, 2016. [Online]. Available: http://www-01.ibm.com/software/data/bigdata/

[36] D. Howe *et al.*, "Big data: The future of biocuration," *Nature*, vol. 455, no. 7209, pp. 47–50, 2008.

[37] C. Lynch, "Big data: How do your data grow?" *Nature*, vol. 455, no. 7209, pp. 28–29, 2008.

[38] M. Hilbert and P. López, "The world's technological capacity to store, communicate, and compute information," *Science*, vol. 332, no. 6025, pp. 60–65, 2011.

[39] J. S. Ward and A. Barker, "Undefined by data: A survey of big data definitions," arXiv:1309.5821, 2013.

[40] A. Sharp, *Dispatch From the Denver Debate*, The Official Twitter Blog, 2012.

[41] F. Michel, "How many photos are uploaded to flickr every day and month?" 2012. [Online]. Available: https://www.flickr.com/photos/franckmichel/6855169886

[42] S. Wang, X. Wang, J. Huang, R. Bie, and X. Cheng, "Analyzing the potential of mobile opportunistic networks for big data applications," *IEEE Netw.*, vol. 29, no. 5, pp. 57–63, Sep./Oct. 2015.

[43] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 29–35, Aug. 2014.

[44] H. Ma, "Internet of Things: Objectives and scientific challenges," *J. Comput. Sci. Technol.*, vol. 26, no. 6, pp. 919–924, 2011.

[45] M. J. Keeling and K. T. Eames, "Networks and epidemic models," *J. Roy. Soc. Interface*, vol. 2, no. 4, pp. 295–307, 2005.

[46] D. Brockmann and D. Helbing, "The hidden geometry of complex, network-driven contagion phenomena," *Science*, vol. 342, no. 6164, pp. 1337–1342, 2013.

[47] C. C. Zou, L. Gao, W. Gong, and D. Towsley, "Monitoring and early warning for Internet worms," in *Proc. 10th ACM Conf. Comput. Commun. Secur.*, Washington, DC, USA, 2003, pp. 190–199.

[48] S. Yu, G. Gu, A. Barnawi, S. Guo, and I. Stojmenovic, "Malware propagation in large-scale networks," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 1, pp. 170–179, Jan. 2015.

[49] H. Luo, J. Luo, Y. Liu, and S. K. Das, "Adaptive data fusion for energy efficient routing in wireless sensor networks," *IEEE Trans. Comput.*, vol. 55, no. 10, pp. 1286–1299, Oct. 2006.

[50] Y. Sun, H. Luo, and S. K. Das, "A trust-based framework for fault-tolerant data aggregation in wireless multimedia sensor networks," *IEEE Trans. Depend. Secur. Comput.*, vol. 9, no. 6, pp. 785–797, Nov./Dec. 2012.

[51] D. Zhao, X. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, 2014, pp. 1213–1221.

[52] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," *ACM SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 29–43, 2003.

[53] J. O. Gutierrez-Garcia and A. Ramirez-Nafarrate, "Collaborative agents for distributed load management in cloud data centers using live migration of virtual machines," *IEEE Trans. Serv. Comput.*, vol. 8, no. 6, pp. 916–929, Nov./Dec. 2015.

[54] Y.-H. Kao, B. Krishnamachari, M.-R. Ra, and F. Bai, "Hermes: Latency optimal task assignment for resource-constrained mobile computing," in *Proc. IEEE INFOCOM*, Hong Kong, 2015, pp. 1894–1902.

[55] 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 Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 3rd Quart., 2014.

[56] Y. Sun and J. Han, "Mining heterogeneous information networks: A structural analysis approach," *ACM SIGKDD Explorat. Newslett.*, vol. 14, no. 2, pp. 20–28, 2012.

[57] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[58] P. Van Mieghem, *Graph Spectra for Complex Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[59] L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. V. Boas, "Characterization of complex networks: A survey of measurements," *Adv. Phys.*, vol. 56, no. 1, pp. 167–242, 2007.

[60] X. Ying, L. Wu, and X. Wu, "A spectrum-based framework for quantifying randomness of social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 12, pp. 1842–1856, Dec. 2011.

[61] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min–max cut algorithm for graph partitioning and data clustering," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, San Jose, CA, USA, 2001, pp. 107–114.

[62] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[63] P. Van Mieghem, J. Omic, and R. Kooij, "Virus spread in networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 1–14, Feb. 2009.

[64] F. D. Sahneh, C. Scoglio, and P. Van Mieghem, "Generalized epidemic mean-field model for spreading processes over multilayer complex networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1609–1620, Oct. 2013.

[65] G. Gürsun, M. Crovella, and I. Matta, "Describing and forecasting video access patterns," in *Proc. IEEE INFOCOM*, Shanghai, China, 2011, pp. 16–20.

[66] M. Wattenhofer, R. Wattenhofer, and Z. Zhu, "The Youtube social network," in *Proc. ICWSM*, Dublin, Ireland, 2012, pp. 354–361.

[67] S. D. Roy, T. Mei, W. Zeng, and S. Li, "Towards cross-domain learning for social video popularity prediction," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1255–1267, Oct. 2013.

[68] Y. Tian, R. A. Hankins, and J. M. Patel, "Efficient aggregation for graph summarization," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Vancouver, BC, Canada, 2008, pp. 567–580.

[69] D. Chakrabarti and C. Faloutsos, "Graph mining: Laws, generators, and algorithms," *ACM Comput. Surv. (CSUR)*, vol. 38, no. 1, Art. no. 2, 2006.

[70] D. Chakrabarti, C. Faloutsos, and Y. Zhan, "Visualization of large networks with min-cut plots, A-plots and R-mat," *Int. J. Human Comput. Stud.*, vol. 65, no. 5, pp. 434–445, 2007.

[71] M. E. J. Newman, "The structure and function of complex networks," *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, 2003.

[72] J. F. Rodrigues, Jr., *et al.*, "Large graph analysis in the gmine system," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 106–118, Jan. 2013.

[73] L. Shi, H. Tong, J. Tang, and C. Lin, "VEGAS: Visual influence graph summarization on citation networks," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 12, pp. 3417–3431, Dec. 2015.

[74] F. Chierichetti *et al.*, "On compressing social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2009, pp. 219–228.

[75] Y. Lim, U. Kang, and C. Faloutsos, "SlashBurn: Graph compression and mining beyond caveman communities," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 3077–3089, Dec. 2014.

[76] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Statistical properties of community structure in large social and information networks," in *Proc. 17th Int. Conf. World Wide Web*, Beijing, China, 2008, pp. 695–704.

[77] P. Boldi and S. Vigna, "The Webgraph framework I: Compression techniques," in *Proc. 13th Int. Conf. World Wide Web*, New York, NY, USA, 2004, pp. 595–602.

[78] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 4, pp. 251–262, 1999.

[79] J.-D. J. Han *et al.*, "Evidence for dynamically organized modularity in the yeast protein–protein interaction network," *Nature*, vol. 430, no. 6995, pp. 88–93, 2004.

[80] S. Hadlak, H.-J. Schulz, and H. Schumann, "In situ exploration of large dynamic networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2334–2343, Dec. 2011.

[81] M. Burch, C. Vehlow, F. Beck, S. Diehl, and D. Weiskopf, "Parallel edge splatting for scalable dynamic graph visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2344–2353, Dec. 2011.

[82] D. Mashima, S. G. Kobourov, and Y. Hu, "Visualizing dynamic data with maps," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 9, pp. 1424–1437, Sep. 2012.

[83] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT press, 2009.

[84] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[85] V. C. Klema and A. J. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Trans. Autom. Control*, vol. 25, no. 2, pp. 164–176, Apr. 1980.

[86] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.

[87] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.

[88] N. K. Logothetis and D. L. Sheinberg, "Visual object recognition," *Annu. Rev. Neurosci.*, vol. 19, no. 1, pp. 577–621, 1996.

[89] S. E. Palmer, "Hierarchical structure in perceptual representation," *Cogn. Psychol.*, vol. 9, no. 4, pp. 441–474, 1977.

[90] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[91] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[92] Y. Zheng and X. Xie, "Learning travel recommendations from user-generated GPS traces," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 1, Art. no. 2, 2011.

[93] J. Shang, Y. Zheng, W. Tong, E. Chang, and Y. Yu, "Inferring gas consumption and pollution emission of vehicles throughout a city," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, New York, NY, USA, 2014, pp. 1027–1036.

[94] S. Z. Li, X. W. Hou, H. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. 2001, pp. I-207–I-212.

[95] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retriev.*, Toronto, ON, Canada, 2003, pp. 267–273.

[96] H. Liu, Z. Wu, X. Li, D. Cai, and T. S. Huang, "Constrained nonnegative matrix factorization for image representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1299–1311, Jul. 2012.

[97] E. Esser, M. Möller, S. Osher, G. Sapiro, and J. Xin, "A convex model for nonnegative matrix factorization and dimensionality reduction on physical space," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3239–3252, Jul. 2012.

[98] L. Kuang *et al.*, "A tensor-based approach for big data representation and dimensionality reduction," *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 3, pp. 280–291, Sep. 2014.

[99] X. Gao, X. Li, J. Feng, and D. Tao, "Shot-based video retrieval with optical flow tensor and HMMs," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 140–147, 2009.

[100] X. Li, S. Lin, S. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 342–352, Apr. 2008.

[101] S. Yan *et al.*, "Discriminant analysis with tensor representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. San Diego, CA, USA, 2005, pp. 526–532.

[102] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1336–1353, Jun. 2013.

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

[104] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear subspace analysis of image ensembles," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Madison, WI, USA, 2003, pp. 93–99.

[105] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear independent components analysis," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. San Diego, CA, USA, 2005, pp. 547–553.

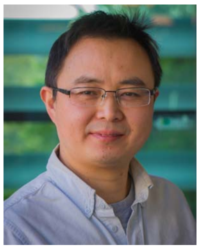
[106] D. Tao, X. Li, W. Hu, S. Maybank, and X. Wu, "Supervised tensor learning," in *Proc. 5th IEEE Int. Conf. Data Min.*, Houston, TX, USA, 2005, p. 8.

[107] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.

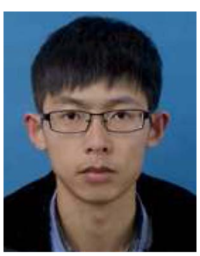
[108] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[109] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.
[110] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1395–1408, Jul. 2009.
[111] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 92–101, Mar. 2008.
[112] S. Li, L. D. Xu, and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and Internet of things," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2177–2186, Nov. 2013.
[113] M. Balazinska *et al.*, "Data management in the worldwide sensor Web," *IEEE Pervasive Comput.*, vol. 6, no. 2, pp. 30–40, Apr./Jun. 2007.
[114] L. Kong *et al.*, "Data loss and reconstruction in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2818–2828, Nov. 2014.
[115] R. B. Cooper, *Introduction to Queueing Theory.* New York, NY, USA: Macmillan, 1972.
[116] S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can we beat DDoS attacks in clouds?" *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2245–2254, Sep. 2014.
[117] H. Khazaei, J. Misic, and V. B. Misic, "Performance analysis of cloud computing centers using M/G/m/m+r queuing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 5, pp. 936–943, May 2012.
[118] K. Salah, K. Elbadawi, and R. Boutaba, "Performance modeling and analysis of network firewalls," *IEEE Trans. Netw. Serv. Manage.*, vol. 9, no. 1, pp. 12–21, Mar. 2012.
[119] M. Liu, W. Dou, S. Yu, and Z. Zhang, "A decentralized cloud firewall framework with resources provisioning cost optimization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 3, pp. 621–631, Mar. 2015.
[120] M. Mitzenmacher, "A brief history of generative models for power law and lognormal distributions," *Internet Math.*, vol. 1, no. 2, pp. 226–251, 2004.
[121] W. Wang, K. Zhu, L. Ying, J. Tan, and L. Zhang, "Map task scheduling in MapReduce with data locality: Throughput and heavy-traffic optimality," *IEEE Trans. Netw.*, vol. 24, no. 1, pp. 190–203, Feb. 2016.
[122] C. Wang, C. Jiang, Y. Liu, X.-Y. Li, and S. Tang, "Aggregation capacity of wireless sensor networks: Extended network case," *IEEE Trans. Comput.*, vol. 63, no. 6, pp. 1351–1364, Jun. 2014.
[123] Z. Li, M. Li, J. Wang, and Z. Cao, "Ubiquitous data collection for mobile users in wireless sensor networks," in *Proc. IEEE INFOCOM*, Shanghai, China, 2011, pp. 2246–2254.
[124] L. He *et al.*, "Evaluating service disciplines for on-demand mobile data collection in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 4, pp. 797–810, Apr. 2014.
[125] S. Chen *et al.*, "When queueing meets coding: Optimal-latency data retrieving scheme in storage clouds," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, 2014, pp. 1042–1050.
[126] A. Thomasian, "Analysis of fork/join and related queueing systems," *ACM Comput. Surv.*, vol. 47, no. 2, pp. 1–71, 2014.
[127] C.-S. Chang, *Performance Guarantees in Communication Networks*. London, U.K.: Springer, 1999.
[128] Y. Jiang and Y. Liu, *Stochastic Network Calculus*. London, U.K.: Springer, 2008.
[129] M. Fidler and Y. Jiang, "Non-asymptotic delay bounds for (k, l) fork-join systems and multi-stage fork-join networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, 2016, pp. 1–9.
[130] M. Alizadeh *et al.*, "pFabric: Minimal near-optimal datacenter transport," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 435–446, 2013.
[131] M. Chowdhury, M. Zaharia, J. Ma, M. I. Jordan, and I. Stoica, "Managing data transfers in computer clusters with orchestra," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 98–109, 2011.
[132] S. Luo *et al.*, "Minimizing average coflow completion time with decentralized scheduling," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., 2015, pp. 307–312.
[133] Y. Zhao *et al.*, "Rapier: Integrating routing and scheduling for coflow-aware data center networks," in *Proc. IEEE INFOCOM*, Hong Kong, 2015, pp. 424–432.
[134] M. Chowdhury and I. Stoica, "Efficient coflow scheduling without prior knowledge," in *Proc. ACM Conf. Spec. Interest Group Data Commun.*, London, U.K., 2015, pp. 393–406.
[135] C.-Y. Hong, M. Caesar, and P. Godfrey, "Finishing flows quickly with preemptive scheduling," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 127–138, 2012.
[136] A. Clauset, C. R. Shalizi, and M. E. Newman, "Power-law distributions in empirical data," *SIAM Rev.*, vol. 51, no. 4, pp. 661–703, 2009.
[137] R. L. Axtell, "Zipf distribution of us firm sizes," *Science*, vol. 293, no. 5536, pp. 1818–1820, 2001.
[138] M. E. J. Newman, *Networks: An Introduction*. Oxford, U.K.: Oxford Univ. Press, 2010.
[139] H. Holm, "A large-scale study of the time required to compromise a computer system," *IEEE Trans. Depend. Secur. Comput.*, vol. 11, no. 1, pp. 2–15, Jan./Feb. 2014.
[140] G. Concas, M. Marchesi, A. Murgia, R. Tonelli, and I. Turnu, "On the distribution of bugs in the eclipse system," *IEEE Trans. Softw. Eng.*, vol. 37, no. 6, pp. 872–877, Nov./Dec. 2011.
[141] M. J. Fischer, D. M. B. Masi, D. Gross, J. Shortle, and P. H. Brill, "Using quantile estimates in simulating Internet queues with Pareto service times," in *Proc. 33nd Conf. Winter Simulat.*, Arlington County, VA, USA, 2001, pp. 477–485.
[142] S. Xu, B. Xu, and D. Peng, "The waiting time distribution of a Pareto service self-similar queuing model for wireless network nodes," in *Proc. Int. Conf. Wireless Commun. Netw. Mobile Comput. (WiCOM)*, Wuhan, China, 2006, pp. 1–3.
[143] R. Rodrfguez-Dagnino, "Some remarks regarding asymptotic packet loss in the Pareto/M/1/K queueing system," *IEEE Commun. Lett.*, vol. 9, no. 10, pp. 927–929, Oct. 2005.
[144] M. E. Newman, "Power laws, Pareto distributions and Zipf's law," *Contemp. Phys.*, vol. 46, no. 5, pp. 323–351, 2005.
[145] T. N. Minh, T. Nam, and D. H. J. Epema, "Parallel workload modeling with realistic characteristics," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2138–2148, Aug. 2014.
[146] K. Wang, Z. Chen, and H. Liu, "Push-based wireless converged networks for massive multimedia content delivery," *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2894–2905, May 2014.
[147] J. Misic and V. Misic, "Probability distribution of spectral hole duration in cognitive networks," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, 2014, pp. 2103–2111.
[148] S. Tang, Y. Dong, and X. Zhang, "Impulse response modeling for underwater wireless optical communication links," *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 226–234, Jan. 2014.
[149] R. Elshawi, O. Batarfi, A. Fayoumi, A. Barnawi, and S. Sakr, "Big graph processing systems: State-of-the-art and open challenges," in *Proc. IEEE 1st Int. Conf. Big Data Comput. Serv. Appl. (BigDataService)*, San Francisco, CA, USA, 2015, pp. 24–33.
[150] R. Chen, X. Weng, B. He, and M. Yang, "Large graph processing in the cloud," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Indianapolis, IN, USA, 2010, pp. 1123–1126.
[151] U. Kang, H. Tong, J. Sun, C.-Y. Lin, and C. Faloutsos, "GBASE: A scalable and general graph management system," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, San Diego, CA, USA, 2011, pp. 1091–1099.
[152] U. Kang and C. Faloutsos, "Big graph mining: Algorithms and discoveries," *ACM SIGKDD Explorat. Newslett.*, vol. 14, no. 2, pp. 29–36, 2013.
[153] G. Malewicz *et al.*, "Pregel: A system for large-scale graph processing," in *Proc. 2010 ACM SIGMOD Int. Conf. Manage. Data*, Indianapolis, IN, USA, 2010, pp. 135–146.
[154] Y. Low *et al.*, "Graphlab: A new framework for parallel machine learning," arXiv:1408.2041, 2014.
[155] B. Shao, H. Wang, and Y. Li, "Trinity: A distributed graph engine on a memory cloud," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, New York, NY, USA, 2013, pp. 505–516.
[156] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput.*, vol. 10. Boston, MA, USA, 2010, p. 10.
[157] R. Li, W. Zhang, Y. Zhao, Z. Zhu, and S. Ji, "Sparsity learning formulations for mining time-varying data," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1411–1423, May 2015.
[158] R. Tibshirani, "Regression shrinkage and selection via the lasso: A retrospective," *J. Roy. Stat. Soc. Series B (Stat. Methodol.)*, vol. 73, no. 3, pp. 273–282, 2011.
[159] K. S. Xu, M. Kliger, and A. O. Hero, III, "Adaptive evolutionary clustering," *Data Min. Knowl. Discov.*, vol. 28, no. 2, pp. 304–336, 2014.
[160] S. Michalak, A. DuBois, D. DuBois, S. V. Wiel, and J. Hogden, "Developing systems for real-time streaming analysis," *J. Comput. Graphical Stat.*, vol. 21, no. 3, pp. 561–580, 2012.
[161] J. Luts, "Real-time semiparametric regression for distributed data sets," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 545–557, Feb. 2015.
[162] H. C. Triandis, *Culture and Social Behavior*. New York, NY, USA: McGraw-Hill, 1994.
[163] W. Bai *et al.*, "Information-agnostic flow scheduling for commodity data centers," in *Proc. 12th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Oakland, CA, USA, 2015, pp. 455–468.
[164] C. Tankard, "Big data security," *Netw. Secur.*, vol. 2012, no. 7, pp. 5–8, 2012.
[165] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 843–859, 2nd Quart., 2013.
[166] C.-J. Chung, P. Khatkar, T. Xing, J. Lee, and D. Huang, "NICE: Network intrusion detection and countermeasure selection in virtual network systems," *IEEE Trans. Depend. Secur. Comput.*, vol. 10, no. 4, pp. 198–211, Jul./Aug. 2013.
[167] M. Ficco and M. Rak, "Stealthy denial of service strategy in cloud computing," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 80–94, Jan./Mar. 2015.

[168] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, “Privacy-preserving data publishing: A survey of recent developments,” *ACM Comput. Surv.*, vol. 42, no. 4, 2010, Art. no. 14.
[169] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, “Privacy aware learning,” *J. ACM*, vol. 61, no. 6, pp. 1–57, 2014.
[170] D. Yang, X. Fang, and G. Xue, “Truthful incentive mechanisms for k-anonymity location privacy,” in *Proc. IEEE INFOCOM*, Turin, Italy, 2013, pp. 2994–3002.
[171] X. Liu, K. Liu, L. Guo, X. Li, and Y. Fang, “A game-theoretic approach for achieving k-anonymity in location based services,” in *Proc. IEEE INFOCOM*, Turin, Italy, 2013, pp. 2985–2993.
[172] Z. Ling *et al.*, “A new cell-counting-based attack against tor,” *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, pp. 1245–1261, Aug. 2012.
[173] S. Yu, “Big privacy: Challenges and opportunities of privacy study in the age of big data,” *IEEE Access*, vol. 4, pp. 2751–2763, 2016.



Shui Yu is currently a Senior Lecturer with the School of Information Technology, Deakin University. He has published two monographs and edited two books on big data, over 150 technical papers, including top journals and conferences, such as the IEEE Transactions on Parallel and Distributed Systems, the IEEE Transactions on Cloud Computing, the IEEE Transactions on Computational Social Systems, the IEEE Transactions on Computers, the IEEE Transactions on Information Forensics and Security, the IEEE Transactions on Mobile Computing, the IEEE Transactions on Knowledge and Data Engineering, the IEEE Transactions on Emerging Topics in Computing, and the IEEE INFOCOM. He initiated the research field of networking for big data in 2013. He has an H-index of 21. He actively serves his research communities in various roles. His research interest includes networking theory, big data, and mathematical modeling. He served as an Associate Editor for the IEEE Transactions on Parallel and Distributed Systems from 2013 to 2015, and is currently serving on the editorial boards of the IEEE Communications Surveys and Tutorials (Exemplary Editor for 2014), the IEEE Access, the IEEE Journal of Internet of Things, the IEEE Communications Letters, the *IEEE Communications Magazine*, and a number of other international journals. He has organized several special issues on big data and cybersecurity. He has served over 70 international conferences as a member of the organizing committee, such as the Publication Chair for the IEEE GLOBECOM 2015 and the IEEE INFOCOM 2016, a TPC Co-Chair for the IEEE BigDataService 2015, the IEEE ITNAC 2015, and an Executive General Chair for ACSW 2017. He was a member of the Deakin University Academic Board from 2015 to 2016, AAAS and ACM, and the Vice Chair of Technical Subcommittee on Big Data Processing, Analytics, and Networking of the IEEE Communication Society.



Meng Liu received the bachelor’s degree from the Software School, Xidian University, and the master’s degree from the Department of Computer Science and Technology, Nanjing University, China, where he is currently pursuing the Ph.D. degree. His research interests include cloud computing, privacy, and security.

Wanchun Dou received the Ph.D. degree in mechanical and electronic engineering from the Nanjing University of Science and Technology, China, in 2001. From April 2001 to December 2002, he was a Post-Doctoral Researcher with the Department of Computer Science and Technology, Nanjing University, China. He is currently a Full Professor with the State Key Laboratory for Novel Software Technology, Nanjing University. From April 2005 to June 2005, and from November 2008 to February 2009, he was with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, as a Visiting Scholar. He has chaired three NSFC projects and published over 60 research papers in international journals and international conferences. His research interests include workflow, cloud computing, and service computing.

Xiting Liu is currently studying with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, China. His research interests mainly include computer networks, big data, privacy, and security.

Sanming Zhou received the Ph.D. degree (with distinction) from the University of Western Australia in 2000. Since then he has been with the School of Mathematics and Statistics, University of Melbourne, Australia, where he is currently a Reader. His research interest spans from pure to applied aspects of discrete mathematics, including algebraic graph theory, combinatorial optimization, random graph processes, and various problems from theoretical computer science. He has published over 90 research papers in high quality international journals in the above areas. He was a recipient of the Kirkman Medal by the International Institute of Combinatorics and Its Applications for his outstanding contributions in 2003. In 2004, he was elected as a fellow of the International Institute of Combinatorics and Its Applications. From 2012 to 2015, he held a prestigious Future Fellowship of the Australian Research Council.