This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

Mobile Big Data: The Fuel for Data-Driven Wireless

Xiang Cheng, *Senior Member, IEEE,* Luoyang Fang, *Student Member, IEEE*, Liuqing Yang, *Fellow, IEEE*, Shuguang Cui, *Fellow, IEEE*

***Abstract*—In the past decade, the smart phone evolution has accelerated the proliferation of the mobile Internet and spurred a new wave of mobile applications, leading to an unprecedented mobile data volume generated from the mobile devices, content servers, and network operators, which are mainly non-structured. In this big data era, such non-structured data fragments are pieced together such that, drastically differing from the traditional practice where services determine and define the data, data is becoming a proactive entity that may drive and even create new services. Compared with the so termed 5V characteristics of generic big data, namely volume, variety, velocity, veracity and value, mobile big data is distinct in its unique multi-dimensional, personalized, multi-sensory, and real-time features. In this survey, we provide in-depth and comprehensive coverage on the features, sources and applications of mobile big data, as well as the current state-of-the-art, challenges and opportunities for research and development in this field, with an emphasis on the user modeling, infrastructure supporting, data management, and knowledge discovery aspects.**

I. Introduction

The smart phone evolution in the past decade has accelerated the proliferation of mobile Internet and spurred a new wave of mobile applications on smart phones. In particular, GPS is becoming part of the default configuration of any smart mobile devices, rendering location information readily available. Even in the lack of exact location information when GPS is not enabled, the coarse location can still be inferred from the network-level data. The location information alone can already enable a great variety of applications to provide personalized services (context-aware recommendation, next location prediction based traffic time estimation, etc.) and to assist public service planning (e.g., traffic flow analysis, transportation management, city zone recognition, etc.). As smart phones are equipped with a variety of sensors, personal behaviors can be further learned and monitored. In addition, mobile operators can also collect a huge amount of data to monitor the technical and transactional aspects of their networks. It has been recently recognized that such data, known as mobile big data, could well be an under-exploited gold mine for almost all societal sectors.

In the past, non-structured data fragments are usually considered as useless byproducts merely to facilitate the proper flow of structured data. Nowadays, the purpose of big data processing is to piece together such data fragments so as to gain insights on user behaviors, and to reveal underlying routines that may potentially lead to much more informed decisions. Drastically differing from the traditional practice where services determine and define the data, in the big data era, data is becoming a proactive entity that may drive and even create new services.

Compared with the so termed 5V characteristics of generic big data, namely volume, variety, velocity, veracity and value, mobile big data is distinct in its unique multi-dimensional, personalized, multi-sensory, and real-time features [1]. Recent research on mobile big data processing has shown its great potential for diverse purposes ranging from improving traffic management, enabling personal and contextual services, to enhance public security, etc. For instance, data driven activity recognition is essential for healthcare applications [2]; the usage pattern of smart phones could be utilized to learn the mental status of users [3]; and the mobile data can provide critical information to facilitate the resource optimization in communications networks (e.g., enhancing paging efficiency, provisioning future data rate , predicting resource needs, etc.).

The unique value of mobile big data comes from its ubiquity and context-richness. It has been evident that mobile Internet not only offers traditional services running on the fixed Internet, but also enables a broad range of new applications that allow the Internet to immerse into almost every aspect of our modernizing society. In fact, the mobile Internet traffic carries a much richer context, which pinpoints the time, location, activity, social relationship, and surrounding environment of mobile users. Consequently, mobile big data research has a multi-disciplinary nature that demands diversified knowledge from mobile communications and signal processing to machine learning and data mining. The research field of mobile big data has been booming quickly in recent years, but is somewhat fragmented. This paper aspires to provide an integrated picture of this emerging field to bridge multiple disciplines and hopefully, to inspire more coherent future research activities.

In this paper, we survey the mobile big data literature following the mobile data life cycle, namely generations, trans-

This work was in part supported by the National Natural Science Foundation of China (NSFC) under grants 61622101, 61571020, 61328102, and 61629101, the Ministry National Key Research and Development Project under Grant 2016YFE0123100, the open research fund of National Mobile Communications Research Laboratory under Grant 2016D03, Southeast University, the National Science Foundation (NSF) under grants CNS-1343189/1343155, DMS-1521746/1622433, AST-1547436, and ECCS-1508051/1659025, the DoD with grant HDTRA1-13-1-0029, and the Shenzhen Fundamental Research Fund under Grant No. KQTD2015033114415450.

X. Cheng is with the State Key Laboratory of Advanced Optical Communication Systems and Networks, School of Electronics Engineering and Computer Science, Peking University, Beijing, China, and also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China (email: xiangcheng@pku.edu.cn).

L. Fang and L. Yang are with Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523, USA (emails: luoyang.fang@colostate.edu, lqyang@engr.colostate.edu).

S. Cui is with the Department of Electrical and Computer Engineering, University of California, Davis, CA, 95616 and also affiliated with Shenzhen Research Institute of Big Data (email: sgcui@ucdavis.edu).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

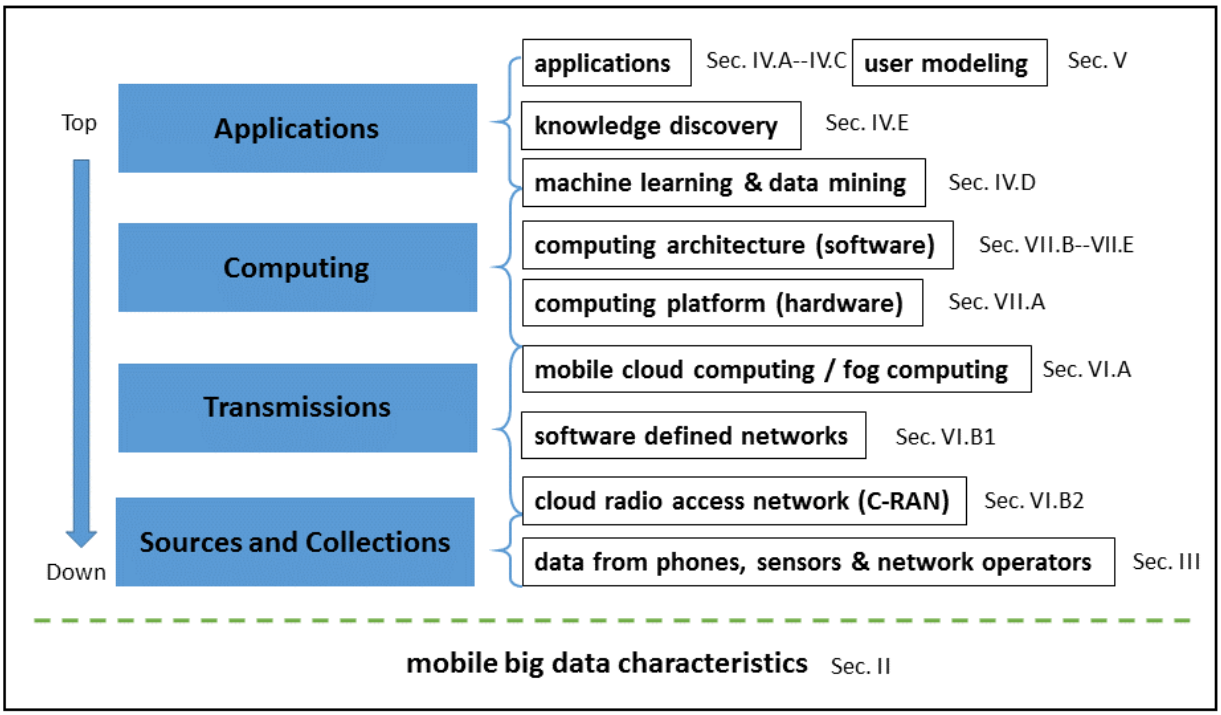


Fig. 1. Topics and Structures of Mobile Big Data

portation, computing and application of mobile big data. The paper is structured to cover multi-discipline topics illustrated in Fig. 1. First, the unique characteristics of mobile big data and its applications are summarized in Sec. II, including the unique "multi-dimensional," "real-time," and "personal" features. The mobile big data generation, including the sources and data collection, is reviewed in Sec. III. Instead of a strictly bottom-up coverage, we first introduce the existing mobile big data applications, the associated methodologies, and the knowledge discovery in Sec. IV. A special application of mobile big data modeling and user behavior profiling is detailed in Sec. V, exploiting the unique "personal" feature of mobile big data. This also serves as a detailed example of methods involved in mobile big data applications. The supporting infrastructures are reviewed in Sec. VI from the various aspects of mobile big data, which facilitate the collection, transportation, computing and application of mobile big data. The newly developed distributed computing architecture for big data is discussed in Sec.VII, which provides an effective tool for mobile big data research, especially on mobile big data mining. Finally, data security and privacy are discussed in the context of mobile big data in Sec. VIII. .

II. Characteristics of Mobile Big Data

As the mobile devices (e.g., smart phones, wearable devices) have become the center of almost everyone's daily life, mining the sheer volume of data from mobile devices has attracted great interests from various research communities, such as data mining, statistics, communications, machine learning, sociology, geography, and so on. This is mainly due to the rich characteristics of mobile big data.

A. "5V" Features

Mobile big data first inherits the "5V" features of generic big data [4], namely volume, velocity, variety, veracity, and value. Though the concept of big data is not precisely defined, its ubiquitous features are well recognized, rendering big data quite different from some simple massive data. The definition of the first "3V" characteristics (volume, velocity, variety) could be dated back to the report by Laney in 2001 [5] and the remaining "2V"s were emphasized in more recent works [6], [7], which are summarized below in the context of mobile big data.

- **Volume.** The *volume* of big data refers to the tremendous size of the data. In the context of mobile data, it is predicted that the mobile data traffic will exceed 15 exabytes per month by 2018 [4].
- **Velocity.** The *velocity* of big data indicates the rapid data generation and streaming. The high penetration of smart devices nowadays, e.g., smart phones, wearable devices, etc., will generate and stream sensed data at an unprecedented speed to facilitate context-aware and personalized applications.
- **Variety.** The *variety* indicates the complexity of mobile big data, which comes from the great heterogeneity in the data types, e.g., multi-sensory data, audio and video footages, etc.
- **Veracity.** The *veracity* suggests the quality of different sources of big data may be inconsistent [6] even in the same domain. Therefore, the data may be noisy, inaccurate and redundant, which should be first cleaned and preprocessed before analysis.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

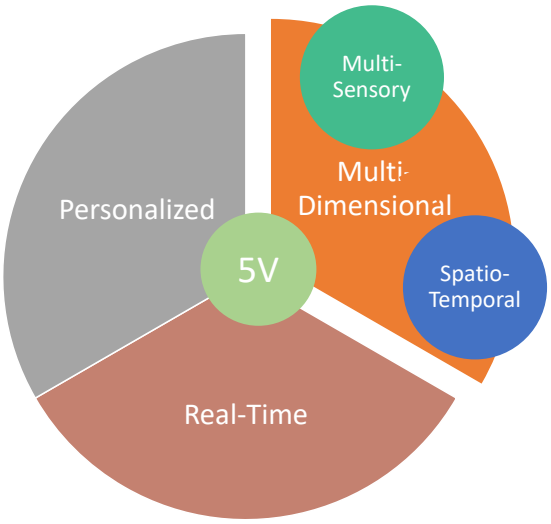


Fig. 2. Distinct characteristics of mobile big data

- **Value.** The *value* of big data was first discussed by Gantz and Reinsel [7], who outlined that the big data technologies hinge upon the economical *value* extraction from the massive volume, high velocity and wide variety of data, with the capability of data analysis and knowledge discovery.

Besides the 5V characteristics as for traditional big data, mobile big data also exhibit some distinct features, which will be introduced in the following subsections.

B. Multi-Dimensional

The multi-dimensional feature is naturally inherent in mobile big data, as it is generated by multiple sensors and tagged with time and geolocation information at varying granularities. Indeed, the *spatio-temporal* and *multi-sensory* features are clearly noticeable in mobile big data. Note that the correlation across dimensions makes this feature distinct from the "variety" feature.

1) Spatio-Temporal: Almost every entry in mobile big data is tagged with a time stamp and certain geolocation information, which enables a great number of new applications. In fact, almost every smart phone is equipped with a GPS receiver, which provides accurate outdoor location information with raw data containing the latitude and longitude. Even when the location service of a smart phone based on GPS is not enabled or not reliable (e.g. when indoor), different granularities of location information can be inferred by other data entries, e.g., service set identifier (SSID) of WiFi access points, cell ID in call detail records (CDR) [8], WiFi signal strength [9]–[11], and even IP addresses [12]–[14].

In particular, the CDR data records the time stamps and approximate location information for all calls and text messages of each user, which are automatically generated by the telecommunication systems. However, the CDR cannot provide the location information when the user is not active. That is, no location information is available between two call records. In [8], Ficek *et al.* proposed a probabilistic model that estimates user locations between their consecutive communication events (calls or text messages), in order to obtain finer trajectories of users from the network cell transition information in CDR.

It is a common consensus that GPS cannot be used indoor. In addition, the location information directly obtained from cell IDs is not sufficiently accurate for certain mobile applications, e.g., location-aware precise mobile advertising. In the literature, localization in indoor scenarios can be achieved by exploiting the WiFi signal strength data. The unpredictability of signal propagation through indoor environments is a major challenge in localization based on WiFi signal strength. Ferris *et al.* in [10] aimed to build a position-conditioned likelihood model for signal strength distributions based on Gaussian process latent variable models, from which the accurate location information can be learned by using simultaneous localization and mapping (SLAM) techniques without any location labels in the training data. In [11], Huang *et al.* improved the computational complexity of the method proposed in [10] from $O(N^3)$ to $O(N^2)$ using GraphSLAM, and relaxed several constraints from [10], e.g., limited predefined shapes (narrow and straight hallways). The accuracy of indoor localization in [11] was claimed to be between 1.75m to 2.18m over an area of 600m^2.

When the location service is not enabled or when users are not willing to share their location information due to privacy concerns even in the outdoor scenarios, the user location information to some degree could be still learned from the available mobile big data to facilitate mobile applications while protecting user privacy. In [14], Long *et al.* proposed an approach to infer the user locations from the hashed user IP addresses at the census block group (CBG) level, where CBG is a geographical unit defined by the United States Census Bureau (USCB) and typically has a population of 600 to 3000.

In addition, the location information is often used to facilitate various recommendation services. However, the raw location information, such as coordinates (longitude and latitude) from GPS receivers, cell IDs from CDRs, or even the indoor location estimated from WiFi signal strength, is meaningless for certain mobile applications (e.g., recommendation services, mobile advertising, etc.), if it is not mapped correctly to what can be understood by human beings. Therefore, tagging the location semantically is critical for many mobile applications. However, it is also challenging, especially when it comes to the extremely dense urban areas, due to the great amounts of location data [15] and the inadequate accuracy of civilian GPS [16]. In [17], Goncalves *et al.* built a crowdsourcing framework termed as *Game of Words* to interact with users for their personalized semantic tagging of locations. The *Game of Words* identifies, filters, and ranks keywords, by which a great number of users characterize a location, such that the semantic location tagging could be adapted to dynamic changes of a location without degradation due to noises and biases as with the single-source data.

2) Multi-Sensory: Almost all smart phones nowadays are equipped with a rich set of embedded sensors [4], e.g., accelerometer, thermometer, compass, gyroscope, GPS signal receiver, ambient light sensor, etc. Such embedded sensors can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

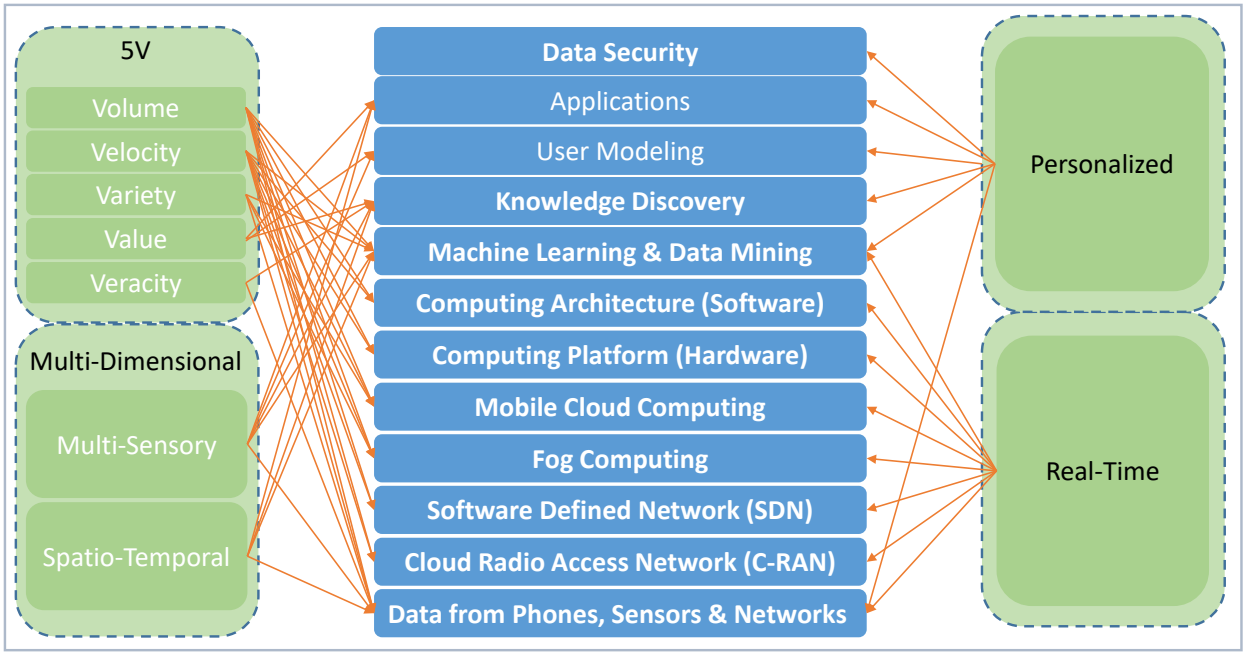


Fig. 3. Distinct characteristics on different topics of mobile big data

provide a tremendous volume of data. For example, one hour of simple personal monitoring (e.g. ECG, HR, accelerometer data, etc.) generates about 14MB of data [18]. With these embedded sensors, context sensing can be performed to facilitate context-aware applications. However, context sensing requires multiple sensors to provide correlated multi-dimensional data simultaneously such that the sensing result could be more accurate. In other words, a single sensor may be of little use semantically in depicting the context of device holders. With smart fusion of data from multiple sensors, more data-driven mobile applications, such as pervasive health computing, activity recognition, context-aware services and so on, could be facilitated by smart devices.

In addition, with the built-in connectivity, smart phones often serve as sensor hubs for wearable sensors [18], e.g., ECG sensors, pedometers, etc. Though the high dimensional data from multiple sensors provides vast possibilities and great potentials for mobile applications, it also inherits some drawbacks from typical sensor data, e.g., incomplete dataset and outliers, due to random sensor failures. This leads to some interesting but challenging problems [19], [20].

C. Real-Time

A typical requirement of analytics over mobile big data is that the location-based and highly personalized information and services need to be delivered to mobile users nearly in real time [21]. For example, the dynamic processing of mobile big data adapted to the context of the device holder requires the environmental sensing data (e.g., temperature, humidity, etc.) almost simultaneously. On the other hand, smart phones highly penetrate the modern lives of users and can serve as a real-time interaction platform between users and applications. That is, mobile big data record the real-time user preferences given different contexts and scenarios. Therefore, the mobile-big-data-driven applications should respond to user requests in near real-time to ensure the quality of experience, especially in certain time-critical applications such as mobile health.

However, the resource limitation of mobile devices in terms of computation, storage, and battery could hardly meet the intensive demands on massive mobile data processing. In this aspect, the mobile cloud computing schemes may provide a solution, which allocate intensive computing demands between the mobile devices and cloud computing devices [4]. Its associated opportunities and challenges will be discussed in Section VI.

D. Personalized

Mobile data directly collected from user devices or mobile networks (e.g., gateways, base stations) contains user identities. Besides the identity information, the mobile data itself is usually highly personalized and linked to user locations and contexts. In fact, the time-stamped geolocation information records the trajectories of users, which exposes their fundamental privacy. For example, the most visited location of a user at night based on GPS is very likely the physical address of the user. However, from the perspective of mobile big data mining, the privacy-sensitive information are inevitably demanded for precisely personalized mobile applications. Therefore, the management and encryption of privacy-sensitive data should be well investigated [22].

In the collection campaign of MDC [23], user privacy was heavily emphasized and protected by careful data collection design. In particular, MDC explicitly guarantees that the data is completely owned by the participants and each individual has the full control rights of their data [24], [25], such as data accessing, data deletion, etc. Also, the identity of users, phone

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

numbers, identifiers of WiFi and Bluetooth nodes are hashed as pseudonyms and the accuracy of location information is mapped to different levels for both privacy protection and data usability. In addition, the data access management for differently authorized privileges should be well designed to regulate the data exposure.

In addition, the trend of mobile big data analytics is not just for analyzing the past or understanding the present, but also for predicting the future [26], which will provide predictive personal services (e.g., smart context-aware personalized services). Therefore, not only the raw data collected are privacy-sensitive, but also will the results mined from mobile big data reveal the daily personal life patterns of users. Therefore, both the data itself and its analysis results should be carefully protected. Otherwise, the availability of data may be in turn jeopardized, for people might end up unwilling to share their data [23].

The semantic extraction of location information could be used to help protect user privacy, as users have options to share their location information through different levels rather than sharing the exact GPS coordinates, e.g., through the levels of city, district, etc. Furthermore, the obfuscation-based techniques may be used to disguise the actual position by providing less accurate or even faked location information [27]. However, if the region level is too coarse, it will jeopardize its usability in mobile applications. In addition, the obfuscation techniques may not be able to protect the privacy of a user, as adversaries may infer the actual location of a user based on their background information. To address this, the location region information can be transformed to different levels, which are carefully designed such that the privacy-sensitive location information may be cloaked without losing too much accuracy. In [27], [28], Damiani *et al.* proposed a privacy-preserving obfuscation environment (PROBE) framework to personalize the protection of sensitive semantic location, based on the privacy profiles generated by users against the privacy attacks of adversaries.

To summarize, mobile big data inherit some traditional features from generic big data but also have several distinct addons. Its multi-dimensional nature from multiple sensors tagged with fine-grained time stamps and geolocation markers provide fuels to accelerate many personalized precise mobile applications. On the other hand, the real-time response requirement of mobile big data applications and privacy-sensitive data management itself will post a great challenge to system design, which will be discussed in Section VI.

III. Data Generation: Source and Collection

Mobile data can be collected from various sources in the mobile network, as depicted in Fig. 4. These data are usually divided into two categories [21]. One category consists of the *app-level* data directly collected by mobile App vendors from mobile phone sensors. As sensor technologies are ubiquitously equipped in smart phones (e.g., GPS, accelerometer, magnetic field sensor, gyroscope, etc.), the phone usually acts as a sensor hub with enriched connectivity for data collection and transmission. The other data category is the *network-level* one traditionally collected by content service providers and mobile operators, which is a vast amount of various mobile service contents, as well as spatio-temporal mobile broadband data about their systems and customers. This type of data records the system status, the service requests, as well as user information (e.g., user ID, location, device type, time stamps, type of service, etc.). In this paper, we mainly focus on the data related to users, from both the app-level and network-level.

In terms of the sources of data collection, the app-level data mainly come from the mobile terminals, whereas the network-level data are usually from the over the top (OTT) servers and the network operators. The raw data collected from these sources is summarized in Fig. 5. Embedded in these raw data is a large amount of valuable information about the users, including user characteristics, habits, preferences, and even motivations and purposes. Harvesting from these raw data, one can construct more useful information such as context, behavior, relationship, etc. Based on these, additional and more implicit information can be further extracted via data mining. Examples include: basic user characteristics (age, gender, race), occupation, group, habit, interest, political opinion, etc. These could then be used in followup data analytics to restore the original context of the related mobile terminal utilization.

Data collection is the process in which data containing user characteristics, preferences, or activities is obtained. The manner in which the data collection is implemented can be classified into implicit and explicit approaches. In the explicit approach, users are prompted to manually provide various information [29]–[32]. While being simple and straightforward, this requires each user to be not only clear about what relevant information he/she is disclosed, but also willing to take time and effort to participate. However, this is usually hard to achieve, as users could be discouraged by such inquiries. On the contrary, the implicit approach does not require manual user intervention and is accomplished without interfering with normal user activities. The implicit approach also facilitates more frequent information updates since explicit user responses are not required in such updates. For these reasons, the implicit approach is more prevalent. Nevertheless, implicitly collected data usually contains quite a lot of redundancy and irrelevant information, which could complicate the followup processing of the data. In the following subsections, we will present the data in terms of app level and network level.

A. The App-Level Data

Data collected from mobile devices may be from either the software side or the hardware side. The hardware-side data includes the device usage information, sensor information, etc. The software-side data includes the application information, the user profile associated with the devices, and the system logs [33]. There have been quite a few projects focusing on the collection of data from the mobile terminals. Reality mining carried out by the MIT Human Dynamics Lab over 9 months in 2004 was among the earliest efforts, where 75 faculty and students with the MIT Media Lab and 25 students at the MIT Sloan business school, participated using 100 Nokia 6600 smart phones [34]. In this experiment, call logs,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

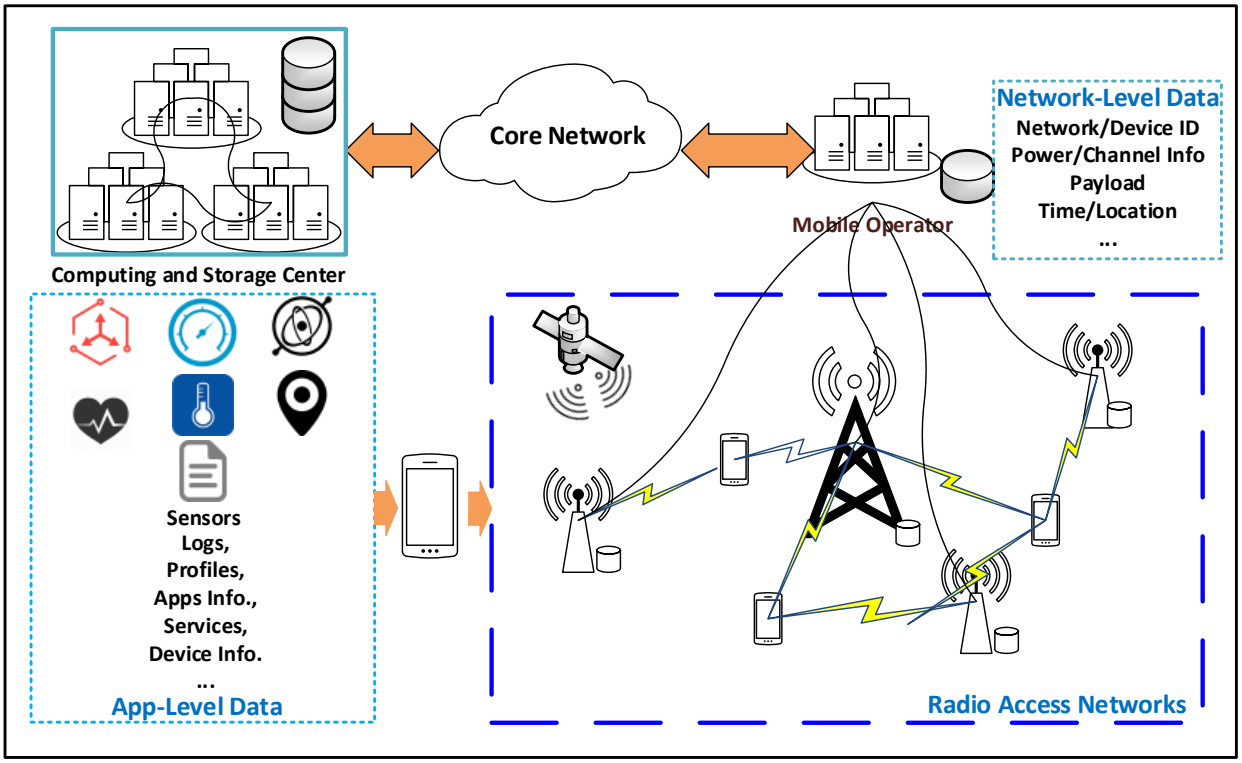


Fig. 4. Data sources

bluetooth devices in proximity, cell tower IDs, phone status (charging or idle), and popular application usage data have been collected. In the more recent Mobile Data Challenge (MDC) by Nokia, 200 volunteers participated using Nokia N95 in the Lake Geneva region from October 2009 to March 2011 [23]. Data collected include calls, short messages, photos, videos, application events, calendar entries, location points, historically connected cell towers, accelerometer samples, Bluetooth observations, historically connected Bluetooth devices, WLAN observations, historically connected WLAN access points and audio samples. Since March 2011, the Device Analyzer experiment at a much larger scale involving $12,500$ Android devices was carried out by the Computer Laboratory at the University of Cambridge [35], [36]. The records of covered countries, phone types, OS versions, device settings, installed applications, system properties, bluetooth devices, WiFi networks, disk storage status, energy and charging status, telephony, data usage, CPU and memory status, alarms, media and contacts, as well as sensors have been collected and analyzed. These campaigns have been summarized in Fig. 6

B. The Network-Level Data

These data are typically collected either at the OTT servers or at the network operator servers. The raw information at the OTT servers consists of a vast amount of texts, user profiles, system logs, audio and visual contents etc. Most of OTT service providers directly interact with end users, rendering network operators pure "pipes," and thus keeping them away from the invaluable data flow.

On the other hand, the radio access network data mainly come from the interactions between mobile terminals and base stations, which involve cell search, synchronization, link establishment, uplink and downlink data transfer, handover, and system information broadcast. These lead to the exchange of a variety of data involving multiple network layers, such as network and device identity, power/carrier/antenna indices, payload and transmission mode, timing information, and location.

Compared with the data from the content service providers and mobile terminal devices, the server data items unique to network operators include: location, address, time, record, flow, URL etc. Among these, "location" contains the locations of the base stations (location area code, LAC), the cells (service area code, SAC) and the routers (routing area code, RAC), from which each individual user's physical position could be uniquely determined, without the assistance of the mobile terminal GPS. "Address" contains the IP addresses of the clients, the servers, and the tunnels, etc. "Time" contains the starting time stamps of user's connections and sessions. Also uniquely accessible by the network operators are the user mobile number (MSISDN) and user device identity (IMEI), from which each individual user's specific device can be determined. These data, being privacy sensitive, are not typically accessible by other sources of data collection, unless voluntarily provided by the users. The latter case, however, could potentially compromise the reliability of collected data depending on the user's true willingness to disclose such data.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

	Data	Parameter
app-level data	Device	Device Type, Device Usage, etc.
	Profile	MSISDN, IMEI, IMSI, User preference, Calendar, Appointment, etc.
	Sensor	Sensing Data, e.g., GPS, Gyroscope, Accelerometer, etc.
	App	Terminal Application Type, Application Usage, etc.
	Service	Service Information, e.g., Bundle Type, Service Charge, etc.
	Log	Terminal Device Log, Server System Log, etc.
network-level data	Time	Connection Starting Time, Session Starting Time, etc.
	Location	Terminal Location, BS Location, Router Location, Cell Location, etc.
	Address	Client IP, Server IP, Client Tunnel IP, Server Tunnel IP, etc.
	URL	Uniform Resource Location, Link Information, Link Content, etc.
	Flow	Uplink traffic, Downlink Traffic, Packet Number, etc.
	Record	Conversation Log, e.g., Conversation Duration, Conversation Time, Conversation Frequency, etc.

Fig. 5. Basic data and parameters.

IV. Applications

Analytics and mining over mobile big data enriched with time and location information will provide great opportunities for new services. The potential applications driven by such mobile user data could be roughly divided into two categories. One is mining on the individual user data to provide personalized services (e.g., context-aware sensing, point of interests, activity recognition, etc). The other is mining on the aggregation of mobile user data to learn and analyze the pattern of human activities, which aims to understand human behaviors in order to help public service planning and city management (e.g., social response monitoring in lieu of social events or disasters, anomaly detection, traffic flow pattern learning, city zone characterization, etc.). However, these two categories are not strictly separable, as some services not only rely on unique patterns mined from individual user data, but also depend on common patterns analyzed from the aggregation of mobile data from multiple users. Details of several typical applications exploiting mobile user data are discussed as follows.

A. Spatio-Temporal Study of Human Beings

The human mobility is of great interest in sociology but has not been thoroughly studied due to the lack of fine-grained geolocation data to record trajectories of individual human beings. The location sequence data collected in the mobile big data can facilitate not only the study and analysis on the human behavior at both individual and aggregated scales, but also mobility prediction based on the behavior pattern revealed by the spatio-temporal dynamics embedded in the data.

1) Fundamental Analysis on Human Mobility: The study and analysis of human mobility is of great interest, especially with the availability of the tremendous volume of spatio-temporal mobile data directly recording people's daily life. As suggested in [37], according to the study on $100,000$ anonymized mobile phone users whose positions were tracked for a six-month period, it was observed that human mobility is highly regularized rather than randomized in both temporal and spatial domains. Additionally, each individual followed a reproducible pattern in terms of characteristic travel distance (exploration) and a significant probability of returning to highly frequented locations (preferential return). The inherent similarity of individual travel patterns has great potentials in public applications driven by human mobilities, such as urban planning, epidemic prevention, and emergency response.

Furthermore, Song *et al.* [38] aimed to quantify the role of randomness in human behavior to answer the question that to what extent the human mobility can be predicted, based on the cell IDs in mobile CDR of $50,000$ individuals each visiting more than 2 locations. With entropy modeling on human behavior randomness, e.g., the number of locations visited, the heterogeneous visiting probabilities of different locations, and the time spent at each location, the predictability of individual mobility could be quantified. As a result, it was concluded in [38] that the human mobility is predictable with up to 93% accuracy based on the user entropy empirically determined from the mobile data and the Fano's inequality.

On the other hand, the privacy protection issue is under significant challenge in the context of mobile big data with spatio-temporal information, as it is shown that the trajectory of an individual is unique and can be easily recognized with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

Project	Time	Organization	Data Collected
Reality Mining http://realitycommons.media.mit.edu/realitymining.html	2004	MIT Human Dynamic Lab	call logs, Bluetooth devices in proximity, cell tow IDs, phone status, popular application usage data
Mobile Data Challenge (MDC) https://www.idiap.ch/dataset/mdc	2009-2011	Nokia	calls, SMS, photos, videos, application events, calendar entries, location points, unique cell towers, accelerometer samples, etc.
Device Analyzer Experiment https://deviceanalyzer.cl.cam.ac.uk/	2011 - ~	Computer Laboratory at the University of Cambridge	covered countries, phone types, OS versions, device settings, installed applications, system properties, Bluetooth devices, WiFi networks, disk storage, energy and charging, telephony, data usage, CPU and memory, alarms, media and contacts

Fig. 6. Summary of Mobile Data Collection Projects

90% accuracy only based on 4 spatio-temporal points [39], regardless of the anonymity in the spatio-temporal mobile data. Intuitively, as the spatio-temporal resolution is getting coarse, the distinct trajectories also become less identifiable. However, this study showed that when the resolution decreases, the trajectory identifiability decays at a rate that is a magnitude of order slower. In other words, even with low-resolution data, user trajectories could still be identified with a relatively high distinction.

2) Location Prediction over Different Time Scales: With semantically meaningful location information, user location prediction is an interesting application of mobile big data. Such interest is further enhanced by the aforementioned high predictability of human mobility we discussed early. Based on these, location prediction has been studied for short [40], medium [41], and long terms [42], respectively. We expect that the full exploitation of location prediction based on human mobility will potentially sprout a great variety of new applications. For each temporal scale of location prediction, the details are discussed as follows.

The *short-term* prediction aims to forecast the destination of users within an hour or less. In [40], Krumm *et al.* proposed a framework called predestination, which attempts to predict the destination based on the GPS trajectory history as the trip progresses over short terms. The destination of a user is recognized based on the length of their sojourn time. In [40], the prediction results were not limited to locations visited in the past but also include the unvisited places, based on the concept of an open-world model with explicit characterization of the model incompleteness. Prediction was carried out by the probabilistic analysis of user destinations and the spatial modeling of a given region. The prediction results could reach median errors of about 2 kilometers based on $3,667$ different GPS trajectories. Instead of predicting the destination, Ziebart *et al.* [43] presented a probabilistic model based on the Markov decision process to characterize observed user behaviors, in order to predict the turns and routes of users during a trip within several minutes.

The *medium-term* mobility prediction intends to first analyze the mobility pattern of human beings on a daily basis and then predict the behavior of an individual within a day or so. Eagle *et al.* [41] modeled the daily behavior of an individual by the weighted aggregation of eigenvectors termed as *eigenbehavior*, a set of principal components generated by the covariance matrix of behavior data collected in the Reality Mining study. However, only 4 locations were considered in [41] for simplicity, namely home, work, elsewhere, and no signal. In fact, the principal component analysis on an individual behavior reveals that the daily behavior of an individual can be characterized at up to 90% accuracy with only 6 principal components. Furthermore, with the principal components learned from the personal dataset, one can predict the behaviors for the rest of the day based on the weights learned from a set of half-day data. In addition, individuals can be clustered into communities based on the similarity of their eigenbehaviors within the society.

Sadilek *et al.* [42] proposed a nonparametric method to extract the significant and effective patterns in human mobility, aiming to predict one's location in the *long-term* future (on the order of months or years). The authors first applied Fourier analysis to recognize significant periodicities in human mobility and then extracted the significant patterns based on the principal component analysis (PCA), on a $32,268$-day

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

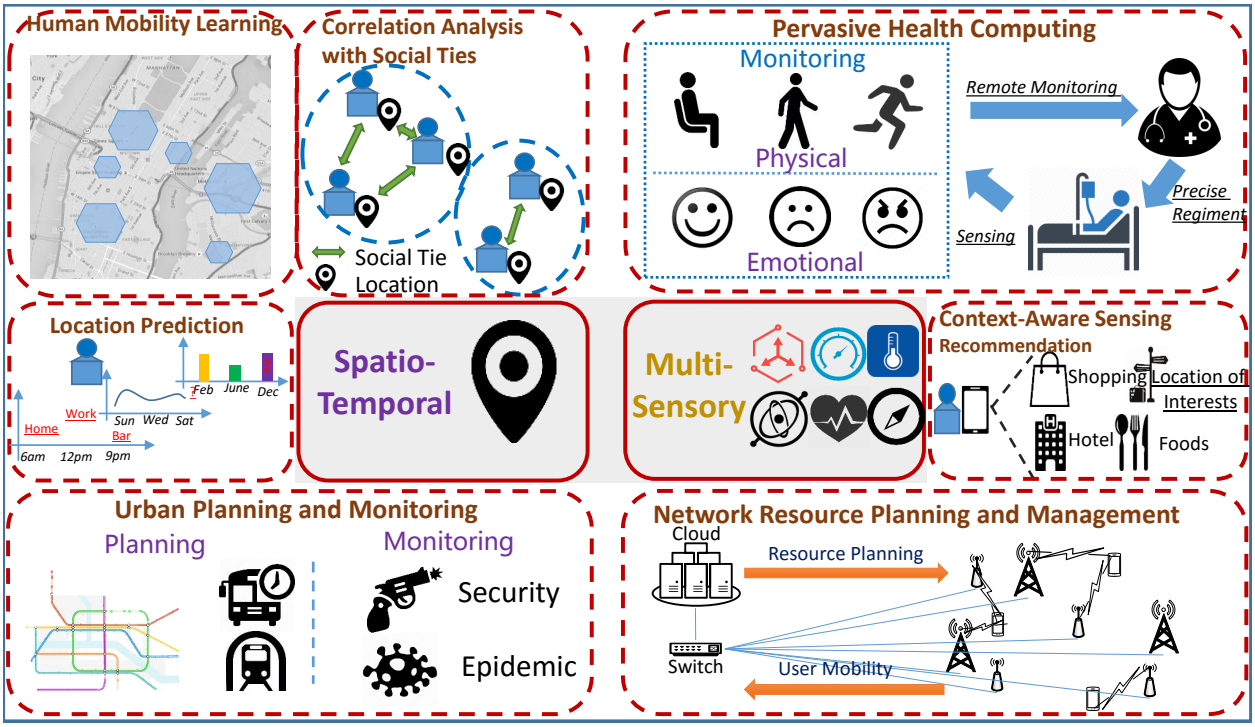


Fig. 7. Applications of Mobile Big Data

worth of GPS location dataset, where the latitude and longitude are represented by a complex number (real and imaginary parts, respectively). In addition, the method was shown to work with very coarse location information, where triangular grids at a 400m resolution are used. The accuracy of the long-term location prediction was claimed to be 80% for up to 80 weeks as shown in [42].

3) Correlation Analysis between Social Tie and Human Mobility: Intuitively, the human mobility is influenced by social ties. As a result, the physical location and social relationships are intrinsically entangled [44]–[47]. Actually, exploitation of correlation between human mobility and social ties is expected to bring two-fold benefits: more regularized location prediction for individuals or social groups with improved accuracy and precision; enhanced identification of social ties and interest groups that in turn facilitates better targeted services.

In [45], a model was introduced by explicitly considering the feedback of mobility on the formation of social ties using data from Twitter, Gowalla, and Brightkite. A model integrating mobility and social interactions has the potential of revealing characteristics of the network topology in terms of total number of the connected components, the distance distribution among connected users, the properties of social overlaps, and the distance-based user clusters. In fact, it was shown in [46] that the exploitation of trajectory correlation between two individuals can improve the human mobility prediction dramatically, where the correlation is characterized by their mutual information.

The geographical distance and its impact on network topologies were mainly studied in [45]. However, the distance between two continuous trajectories may not clearly capture the social tie between them, as multiple different routes in an urban area could reach the same location. Therefore, Toole *et al.* in [47] defined the mobility as discrete visits to places at different time stamps and proposed explicit measures on the patterns of both mobility and social behaviors, in order to quantify the correlation between two individuals, based on which the mobility similarity could be used for semantic social relationship classification, e.g., as coworkers, family members, etc. The mobility similarity between two people is quantified based on their mobility traces of specific times on a given day, which essentially amounts to the calculation of the angle between two vectors representing the CDR-recorded trajectories of two users. In addition, the study also reveals that the temporal correlation inherent in user trajectories plays an important role in the classification of social relationships. For example, the mobility similarity during daytime on weekdays strongly indicates the relationship of coworkers. It also suggests that human mobility is far more similar to each other when two people have certain social relationships than when the two are strangers.

4) Context-Aware Sensing and Recommendation: Context-aware recommendation is an immediate application of mobile big data embedded with spatio-temporal information. It aims at providing precise personal recommendation to users, given the context of a user (i.e., location, time stamp, velocity, acceleration, etc.), as well as the user's personal preferences, where the location information is the fundamental information for context-aware services. Context-aware recommendation is essentially the interaction between the user preferences acquired from his/her behavior history and the available options under a certain context in terms of time, space, environment etc. [48]. Not surprisingly, nearly all context-aware services require certain location information. The integration of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

actual location, the inferred points of interest, the predicted location based on personal location and/or social relationships, and the inferred/refined social roles based on location information, as we discussed before, will undoubtedly enhance the consumer experience of context-aware recommendations. In addition, the location prediction also plays an important role in context-aware recommendations [16].

The common preference among multiple users is important for precise recommendation, as the users in the same group given a specific context have a high probability of sharing some common interests. Furthermore, to mine the context-aware preferences from the context-rich mobile big data is challenging, as the data from one user or a limited number of users might not be sufficient for accurate preference learning. In [49], it was proposed to first derive common context-aware preferences from the data of multiple users, and then to characterise individual user preferences via a probabilistic distribution of these common preferences.

In addition, the social roles of users could be studied based on the abstracted characterization of shared preferences or behavior patterns among multiple users [50], which could be utilized for recommendation or advertisement based on similar social roles. The benefits of social-role based recommendation is more precise and more reliable, as the cross recommendation among users with the same social role can be applied due to the social nature of human beings. In fact, the common preferences among users are not the only factor that influences the choice of a user. In [51], the distance (how far between a candidate restaurant and the user) combined with the common preferences (recorded by the ranks of candidate restaurants) was considered to estimate the likelihood that a user will choose a particular candidate restaurant nearby.

B. Pervasive Health Computing

With the multi-sensor data from wearable devices or smart phones, mobile health (mHealth) is a promising application. Such data from sensors can passively and unobtrusively collect the necessary information for health monitoring. In addition, smart phones with rich connectivity can provide a platform for active data collection in many scenarios, such as facial expression capturing with phone cameras and patient audio clip recording by microphones.

Actually, physical activity monitoring is an intuitive application of health monitoring [2] with such a rich set of sensors. For instance, fall detection coupled with an alert system [52] could be implemented to detect the fall and alert the authorities at the same time. In [53], Wu *et al.* collected the data of the accelerometer and the gyroscope for 16 participants on 13 activities, such as sitting, walking, jogging, and going upstairs and downstairs. Overall, the results of monitoring are claimed to achieve very good accuracies: $52.3\% - 79.4\%$ for up-and-down-stair walking, 91.7% for jogging, $90.1\% - 94.1\%$ for walking on the ground, and 100% for sitting.

In fact, mobile sensors equipped in smart devices can help doctors monitor their patients remotely and frequently. The sensing results will further help doctors customize the personalized medical treatment for each patient. In [54], Sharma *et al.* proposed a framework to help with monitoring Parkinson patients via both active and passive approaches, to understand the daily activities and manage the complex medication regimens personalized to individual needs. In particular, passive monitoring relies on accelerometers as well as gyroscopes to monitor motor aspects of Parkinson patients, such as walking, falls, balance, tremor, and so on, while active monitoring relies on the collection of contextual data, such as speech, facial tremors, etc., by interacting with patients.

Although physical activity monitoring can be considered as a modernized integration of biomedical sensors into personal mobile communication devices, the mental health monitoring hinges more upon context sensing and the development of emotion learning [55]. Based on where we have been, with whom we communicate, what applications we use, and how we use our mobile devices, various learning algorithms can be developed to exploit these mobile phone sensor values. These learning models can be adapted to predict a mobile user's moods, emotions, cognitive/motivational states, activities, environmental contexts, and social contexts.

In [3], LiKamWa *et al.* proposed a mood monitor based on the logged data collected from 32 participants over two months. The authors showed that by analyzing the communication history and application usage patterns, a user's daily mood curve could be statistically inferred with an initial accuracy of 66%, which gradually improves to an accuracy of 93% after a two-month personalized training period. High-quality cameras equipped in smart phones [55] could easily capture our facial expressions indicating emotions and moods, and the new development on emotion learning techniques will further facilitate mobile emotion monitoring. In [56], the smile intensity mined from facial expressions was studied to help machines understand the emotion of human beings, by tracking the changes of facial muscles leading to a specific expression.

Besides psychological monitoring and care for patients, such information can also be employed for a vast array of health/mood related applications. For example, video and music recommendation, context-aware advertisement, and personalized social networking would all significantly benefit from these.

C. Public Service Applications

1) Urban-Related Services: Besides personal applications and services stated above, the aggregation of mobile big data could provide a great tool to represent the big picture of human being social behaviors, e.g., human mobility, or social response and propagation of events, diseases, and disasters. In other words, the mobile big data can help with understanding how the individual dynamics shape the structure of cities, which in turn assists better urban planning and public service planning.

For instance, mobile big data can help reveal regions of different functionalities in urban areas [57], [58] in terms of the aggregation patterns of human mobility. In [57], Grauwin *et al.* discovered a universal structure of cities in terms of general functionalities, based on the analysis of CDRs in three major cities. Actually, the fine-grained functionalities of a region in the city may be overlapping. In [58], Yuan *et al.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

segmented a city into disjointed regions according to major roads and represent each region by a probabilistic distribution on a set of functions, which are learned via a latent variable model (topic-based inference model) with two 3-month GPS trajectory datasets.

Based on the urban study, urban planning (e.g., public transportation planning [59]) can be effectively designed. Traffic patterns can be inferred and different traffic zones can be determined based on the mobile big data, such as CDRs [60], [61]. In [60], Dong *et al.* first semantically extracted and tagged the origins and destinations of traffic commuters from the CDR and utilize mobile phones as well as base stations as sensors to measure the traffic trend in the city. The distribution of travel times learned from mobile device sensors (e.g., accelerators) and GPS trajectories can help monitor the traffic condition and provide the fastest route information for each individual in the area based on the current traffic status [61]. In addition, the social and pathological response and propagation of events, diseases, and disasters can be studied based on the current trends of social networks, physical and emotional sensing, and call data records, which helps improve public security, as well as emergency response and recovery [62], [63].

In fact, urban planning roots in the understanding over the aggregate human mobility in an urban area. The most fundamental problem for urban planning or even network resource management (discussed in next subsection) is the crowd flow analysis in a city. The crowd flow monitoring and prediction could provide a solid foundation for urban sensing and planning. Recent work [64] on crowd flow prediction utilizes the recent development of neural networks (deep learning), which will be discussed in Subsection IV-D. Overall, the crowd flow analysis based on mobile big data is still an open problem.

2) Communication Network Resource Planning: Naturally, based on the aforementioned features, mobile big data provides excellent prior information to feed the the resource allocation and optimization algorithms that are driving the communication networks. In particular, mobile big data can be used to extract, model, and predict the patterns of mobile traffic, which has great potentials towards more efficient network planning and resource management. With innovative development of the cloud radio access network (C-RAN), centralized management of multiple radio access points may be capable of adapting the allocation of network resources according to the dynamics of mobile users, by jointly exploiting the mobile big data that is collected by system operators simultaneously over multiple points [65].

Mobile big data can also be used to gain individual insights such as demographic attributes, mobility patterns, personal preferences, and instant context. Such insights can be used to optimize personal content delivery, contextual services, and mobile advertisement. These knowledge not only can improve the network planning from the traditional perspective in terms of access point planning and flexible radio resource management, but also will be essential for emerging applications such as adaptive content distribution by providing content consumption cartography [66]. In fact, the specific problem formulation on the data-driven communication network resource planning, management, and scheduling is still an open issue, which requires the expertise from both the data mining and communication communities.

D. Methodology

As stated previously, machine learning and data mining (MLDM) tools are critical for mobile big data applications. In fact, MLDM is an interdisciplinary area, crossing statistics, mathematics, computing etc., which are traditionally applied to speech recognition, computer vision, natural language processing and so on. In this subsection, we will review some important concepts of machine learning on mobile big data applications and discuss some future trends in the context of mobile big data.

1) Representation: Generally, the raw data cannot be directly fed to a learning machine, as it may contain redundancy, noise, outliers, etc., which will negatively impact the learning performance. Hence, the data should be first preprocessed prior to the learning process. Actually, the data cleaning alone may not be sufficient, as the raw data sometimes may not even render the features needed for a particular application. Hence, data representation is vital in each machine learning application. This is termed as feature engineering. In fact, a large portion of endeavors in machine learning are devoted to the feature engineering.

For example, locations in mobile big data are the most common yet very critical information. The longitude and latitude information from GPS locations are the utmost raw data, which are samples of the continuous geospace. The CDR contains the discrete locations of the base stations and records such locations corresponding to a Cell ID. The GPS information could be either used directly in its original form without any further processing, as in [36] [42], [57], or processed to provide more informative or convenient features. For example, the number of distinct locations as well as the sojourn time spent at a specific location could be derived from the raw GPS data [38]. On one hand, good features extracted from the raw data could enhance the learning performance of learners, thus making information extraction easier in predictors or capturing the posterior distribution of the underlying explanatory more precisely in probabilistic modeling. On the other hand, if not executed appropriately, feature extraction may lead to detail compromise and information loss from the original data [67]. Hence, feature engineering is indeed a fine art on its own.

Though important, feature engineering is quite labor-intensive in the development of MLDM applications. The future trend is to make feature engineering more automatic rather than manual. This could be fulfilled by a special type of learning [68], namely "representation learning." Representation learning enables MLDM to largely bypass manual feature engineering based on human intuition and prior knowledge, especially under the context of variety inherited in big data. The representation learning has been studied and applied in speech recognition, object recognition, and natural language processing. Due to the distinct multi-sensory and spatial-temporal characteristics in mobile big data, it may be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

overwhelming to extract features manually from such high-dimensional data. Hence, representation learning, may come handy in helping formulate a good representation in order to achieve good MLDM performance on mobile big data.

2) Models: MLDM models could be categorized into two types, namely the descriptive and the predictive. The descriptive type is aimed to illustrate the dependency or relationship among the data, while the predictive type predicts these based on the learned function from labeled data. Depending on whether the data is labeled, the categorization into descriptive and predictive types could also be termed as unsupervised and supervised learning, respectively. The specific models of MLDM used in mobile big data applications are discussed as follows.

Descriptive Methods (Unsupervised)

- Clustering or Segmentation. Clustering is utilized to categorize data into several groups based on the similarities within the group. For example, the type of land use in a city (city functionalities) is segmented using a clustering technique (k-means) on the CDR data in [57].
- Principal Component Analysis (PCA). The PCA is originated from eigen analysis in matrix theory. The PCA is typically applied to learn significant features while reducing the noise or error nested in the data. In [41], daily human mobility behaviors are represented by behavior features, in which eigenvectors (eigenbehaviors) are generated by PCA from the discrete location data in CDR. Then, the daily human mobility behavior is represented as weighted superimposition of the eigenbaviors, which could be applied in mobility prediction.
- Probabilistic Graphical Model (PGM). This model can be used to characterize probabilistic dependencies among different features or even among entries of the raw data with some prior knowledge or assumptions. The captured dependencies could be characterized by graphs (directed or undirected). For example, generative probabilistic models are originally applied to learn latent topics underlying the text corpora, where words in the corpora are assumed to be generated according to a probability distribution of latent topics. Along with similar assumptions, such a topic model is applied to learn city functionalities in [58], in which popularity density reviewed by the mobile big data is assumed to be generated by city functionalities.

Predictive Methods (Supervised)

- Regression and Classification. Regression is aimed to build a function based on the training dataset to generate continuous valued outputs; whereas, classification utilizes the labeled historical dataset to build a classifier, to predict a categorized output given a set of features. The classifier could be linear (e.g., linear discriminant analysis) or non-linear (e.g., support vector machine (SVM)). In the mobile big data context, activity recognition or context-aware related applications based on multi-sensory data from smart phones [52], [53] are typical classification problems.

Besides these two types of categories , "reinforcement learning" has also been proposed and developed in the past years, for which the training data comes from the interaction between the learning machine and the feedback of its environment [69]. In fact, thousands of MLDM models have been presented in the literature and more are published every year. However, to choose an appropriate model that could successfully characterize a specific problem with mobile big data needs significant efforts. In fact, three critical components in a machine learning application should be carefully investigated in this new context: representation, evaluation and optimization [70]. In addition, other special challenges pertinent to MLDM should also be studied. One of the challenges in MLDM is the overfitting problem. The overfitting problem arises when the model (generally a predictor or classifier) characterize too much noise from the training dataset rather than capturing the real merits in data. This will lead to poor performance in validation tests, but could be potentially mitigated by exploiting the large size of data set in mobile big data [70]. The overfitting problem is also critical in the neural network, an important tool that mimics how human brain organizes neurons and learns the environments. Until 2006, the neural network research has been stuck with the overfitting and the high dimensional training issues. "Deep learning" [71], [72] was proposed as a neural network in 2006 with multiple hidden layers (and thus deep). Pretraining via unsupervised learning on each layer of a neural network, deep learning has been successfully applied to several fields. The success of deep learning roots in the progress of computing capabilities as well as the tremendous size of training dataset. In fact, such trends may also be extended to mobile big data analytics. It has been demonstrated in [73] that the deep neural network could achieve better performance than other MLDM techniques (e.g., random forests) in context-aware activity recognition. In addition, unsupervised learning based on the deep neural network structure is attracting significant attention, e.g., autoencoder [72] or generative adversarial networks [74], which could learn the inner feature of data without labels. Such new techniques may potentially facilitate many novel applications, which remains as an active research direction.

E. Knowledge Discovery

Nearly every application or service of mobile big data will be based on the mapping of the raw data to some useful information, and then from the discovered information to intelligent decision. In such, one may be prone to the traps inherent in the complicated knowledge extraction from raw mobile big data, which will in turn lead to false knowledge discovery. The prevention from false knowledge discovery requires deep understanding of application, along with careful verification and validation [75].

Moreover, the conflict between the real-time response requirement of mobile big data applications and extremely large data volumes coupled with velocities not only places great challenges on the infrastructure supporting these applications, but also poses challenges to the algorithm design in mobile big data mining [18]. Such algorithms should be scalable and adaptable to dynamic mobile user environments. In addition,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

how to extract and select features from the multi-sensor mobile big data according to the specific application/service requirement is of great importance, while maintaining reasonable computation complexity in data collection and data processing.

In summary, mobile big data with unique temporal-spatio features will have unprecedented potentials in a great variety of applications for both personal and public services.

V. User Modeling

Most of the aforementioned applications of mobile big data are pertinent to personalized customization of either individual users or user groups. This inevitably necessitates the understanding of user interests to specific information, their behavior traits, and user tendencies during a given time period. Such concepts of user profiling can be dated back to [76]–[80], which refer to the acquisition of each user's basic information, interest and behavior, together with the establishment of a description file [81]. Such user modeling can be regarded as a data preprocessing procedure in order to extract important features from mobile big data to profile each mobile user, which could be further fed to learning machines and data miners to facilitate personalized applications and services.

Although the objective of user modeling is to facilitate precise and dynamic personalized services, description of user interests is not the whole story. Lying in the core of a personalized information service system, computability is another basic requirement in user modeling. In other words, user modeling is an algorithmic and structured user description, which abstracts and extracts computable user models from user interests and behaviors, such as browsed content, browsing behavior, surfing context, and geographic location. In addition, the derived models should be able to evolve with user interests that may vary dynamically. Next, we will overview mobile user modeling from the perspective of the data source. We will see that with different sources, the type, format and application of the data also vary dramatically.

A. With Data from OTT Servers

With OTT servers, we could directly capture the user browsing information. With user interests and preferences being the key concerns of content and service providers, it is natural to develop user models based on user browsing behaviors readily available at OTT servers, such that content/service recommendations can be provided to users and new content services can be developed.

The raw information at the OTT servers includes texts, user profiles, system logs, and URLs, etc. Due to the large quantity of such data, it is not practical to perform detailed analysis on each record, and pre-filtering and pre-processing are often necessary to form somewhat compressed records in order to facilitate the computability of the ensuing data processing and analytics. In particular, the raw HTTP inquiry records are usually filtered to expunge information that are not closely related to the core accessed content. Examples of typically expunged information include graphics, animations, and scripts. The OTT server then combines the filtered records from the same IP address during a short time period to form an approximation of the actual user browsing behavior. In the mean time, as the actual usefulness of the URL access data in indicating user preferences lies in its represented information type, the Open Directory Project (ODP) was devised [82], which is a web directory of Internet resources hierarchically arranged by subject and provides an open framework establishing an ontology of URLs. Using this tool, the user browsing record can be interpreted as the corresponding website type, whose statistics can then lead to user preference profiling.

One of the most important information that can be obtained from the OTT server data is the interaction between users and webpages, i.e., the access rate of each webpage and its corresponding user sojourn time. With this aspect, the mobile user modeling based on the OTT server data is essentially web usage mining [83]–[85], in which the interactions between users and webpages are analyzed and modeled in order to infer user behaviors and preference models. As the main purpose of OTT servers in user modeling is content and service recommendations, which are more user-oriented and personalized, user modeling here is often jointly operated with recommendation systems.

The overall procedure consists of two aspects: data preparation and conversion vs. analysis and modeling. In the data preparation and conversion process, one needs to merge and abstract the data obtained from the servers. One basic abstraction of contents or services is to abstract the raw server data into a pageview, which typically contains the ID (usually a URL), Type (service provided, e.g., information vs. merchandise), and Content (keywords, subject, etc.). One basic user abstraction is to abstract a user visit into a so-termed "Session". Following data preparation and conversion is analysis and modeling, for which some currently adopted methods are listed below:

- **Clustering**. In data preparation, one essentially obtains the interactions among users and contents. Hence the classification methods are either user-oriented or content-oriented. Most commonly adopted algorithms include K-Means [86], hierarchical clustering [87], and expectation-maximization (EM) [88].
- **Association Rule Discovery [89]–[92]**. In [93], clustering and association rule discovery methods are combined, where the user browsing behaviors are first clustered using Relational Fuzzy Subtractive Clustering (RFSC), and the recommendation results are then obtained by applying association rule discovery within each cluster.
- **Latent Variable Models (LVMs)**. Statistical models are used to reveal the underlying relations and structures embedded in the data. Most commonly adopted models are Factor Analysis (FA) models [94] and Finite Mixture Models (FMM) [95].

From the above, it is evident that, as content and service providers, the OTT servers can conveniently establish user interest and preference models from the captured user browsing information, to address their key concerns of personalized content/service recommendations, renovations, and creations. However, user models established solely based on the OTT server data lack the direct information of the user locations,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

device/app utilization habits, and call/text information, etc. This renders joint processing over data from multiple sources were meaningful and necessary.

B. With Data from Mobile Devices

Compared with the data from OTT servers, the device system log and sensor information usually lead to more intimate information about the user. Proper interpretation and analysis of these raw data can reveal the user app utilization habits, device utilization routines, background environments and even user physical and psychological status. Several example applications along this line have been introduced in Section IV. The typical user modeling procedure often contains the following steps:

- **Feature Construction and Extraction**. User feature vectors are constructed from the features extracted from the raw data. Evidently, the selection of features is critically important here, and could vary significantly in different scenarios. For example, [96] considered global and partial user features; [97] examined the inclusion of device utilization behaviors and user locations selected as features, whereas cross validation was adopted in [98] for feature selection.
- **Data Cleaning**. Frequently, missing data needs to be added and feature vectors need to be regularized, leading to dimension reduction. For example, F-Test, Relief, Principal Component Analysis (PCA) and kernel PCA were discussed and applied in [96].
- **Model Establishment**. In [96], the Support Vector Machine (SVM) classifier was employed, and a decision tree was used to fuse results obtained from different models in [98].
- **Prediction and Recommendation**. In this step, user statistical characteristics based on gender, age, marital status, employment, household size, and app utilization habits can all be incorporated into the base of prediction and recommendation [96], [98], [99].

Although data from mobile devices often directly reveal authentic and personal user information, retrieval of such data necessitates access to the device system log and sensor information, which are either privacy-sensitive or only available to certain mobile software developers/providers. Hence, aggregation of such data from a diversity of terminal applications, a user pool of different types and a variety of devices may be challenging.

C. With Data from Network Operators

In communications networks, all user network access behaviors are recorded, leading to a comprehensive user pool, diverse user behaviors, and multidimensional user data, as detailed in Section III. In addition, network operators could further analyze the signaling messages from the communication network gateway to acquire customer numbers, terminal IDs, communication behaviors, texting behaviors, surfing times, browsed contents, operating systems, access point names, traffic flows, and precise locations.

All these facilitate in-depth user modeling involving usage preference profiling, online routine prediction, location tracking, and device characterization, just to name a few. Compared to the data from OTT servers, the data from network operators allow for reconstruction of the physical settings of the user access, such as the corresponding time, location and context. Compared to the data from mobile devices, the data from network operators are from all users across the network, from different types of devices, and for various terminal applications, making large-scale data mining and statistical modeling possible. With such data, it is possible to not only make content/service recommendations based on user interest modeling alone as with the OTT server data, but also fine-tune such recommendations based on user mobility traces or activity routines. If further coupled with the user mood analysis from call/text records and device/app utilization, the recommendations could be further adapted to the user psychological status.

As a result, mining the mobile user data from network operators could reveal not only what the users want, but also where, when, and how the users want. The main research issue is then how to make customized and optimized selection of user features for specific applications. For example, in [100], user features were constructed based on user behavior, utilization, location characterization, etc., and K-means classification was then applied to enable effective router selection by administrators. A number of research results also utilized the location information together with the context of user relationship to infer the user trajectory [101]–[103], or user interests [103], [104]. The user trajectories, together with traffic data, can facilitate realistic network traffic flow simulations of the entire network [104], which may even assist transportation administration and optimization [103]. In some research efforts, the communication record is used to infer the user interests and groups. For instance, communication logs and feedbacks at the customer service centers were used to help operators construct better customer satisfaction models in [105]. In [106], communication behaviors and relationships among users were used to infer the features of their neighbors in the network.

To summarize, in terms of both the variety and accessibility, data from network operators provides the greatest potential for next-generation precision personalized content/merchandise services, thanks to the comprehensiveness, abundance, multi-dimensionality, and continuity in both space and time. In the longer term, its integration with the more detailed content information at the OTT servers and more intimate user information at mobile devices, will open new doors towards fine-grained mobile service personalization. To bring all these together, an interweaving supporting infrastructure will become a necessity, as we will discuss in next sections. The need of new paradigms supporting large-scale mobile data collection, processing and sharing brings not only challenges but also great research opportunities, which will be discussed in the next section.

VI. Infrastructure for Data Transmission

The collection, transmission, and computing of mobile big

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

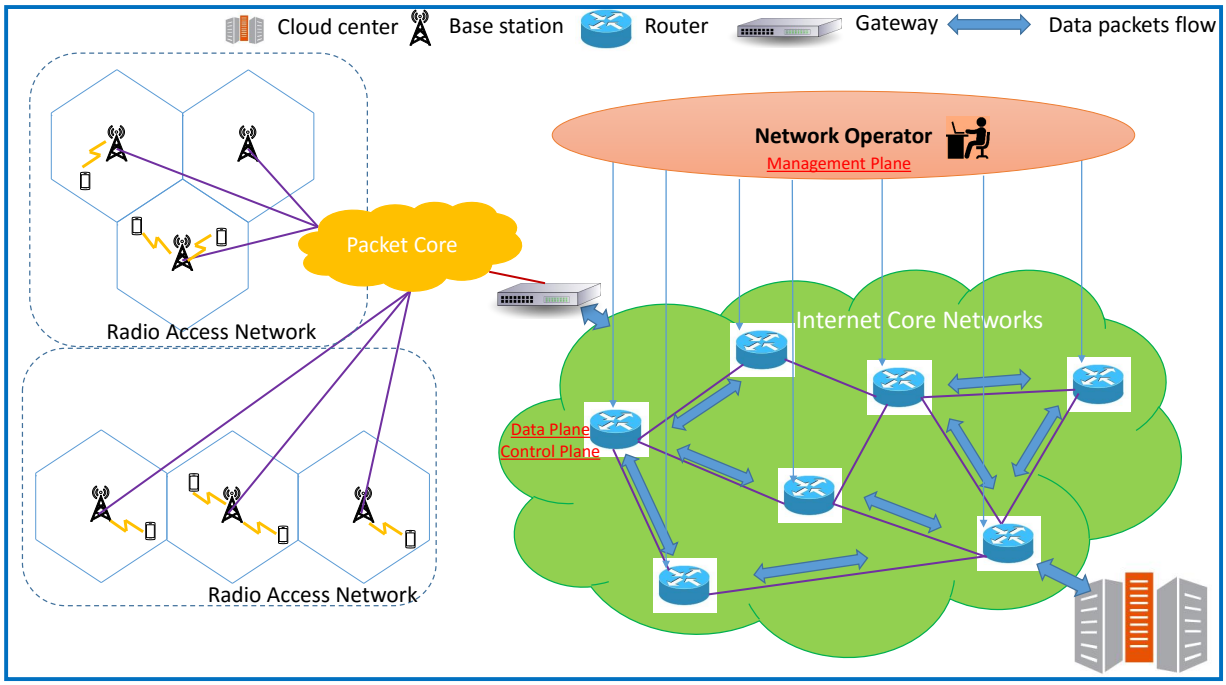


Fig. 8. Mobile Big Data Supporting Infrastructure

data require the support of communication, networking and computing infrastructure. Due to the special characteristics of mobile big data, the communications and networking infrastructure urges a revolutionary overhaul. For example, the (near) real-time response demanded by some mobile big data driven applications is hardly satisfied by the existing infrastructure. In this section, we survey the potential technologies on communications, transmissions and computing in the context of mobile big data.

Research challenges on the infrastructure supporting mobile big data are always entangled with the tradeoff between centralization and distribution of resource management and system design. Specifically, centralization brings efficiency and convenience to the system management and coordination, but falls short in terms of scalability. On the other hand, distribution usually leads to improved scalability, but lacks the easiness on global system management and coordination. Hence, the issue of how to design the system to support mobile big data collection, processing and sharing, considering the tradeoff between centralization and distribution, is always of great interest, which will be discussed in the following subsections.

A. Computing Infrastructure

1) Mobile Cloud Computing: The concept of centralized mobile cloud computing (MCC) [4], [107] (Fig. 9) is proposed to solve the problem of mobile big data processing, by integrating mobile sensing and cloud computing. The intensive computing workload and high-volume data storage demand of mobile big data processing are loaded to the cloud via certain access and backhaul networks.

With the idea of MCC, the bottleneck of mobile big data processing is shifted to the communication between the mobile devices and the cloud. The involved access and backhaul connections should be able to handle massive data transmissions due to the tremendous volume of mobile big data, as well as massive simultaneous device connection requests.

There are some major challenges to apply MCC for mobile big data processing. First, the current radio access networks may not be able to meet the intensive future needs of mobile big data transmissions. In addition, the MCC needs to adapt to the randomly varying communication quality, low security and high probabilities of signal interception [108]. Secondly, the latency due to access and backhaul networks is a vital challenge [109] for mobile cloud computing, especially when interactions between mobile terminals and the cloud are required in real time. In addition, the degrading communication quality will be intensified by the high latency of the backhaul networks, and such latency is difficult to control in traditional networks, as routers and switches in the traditional computer networks are locally operated and controlled. Hence, how to reduce the high transmission latency in the context of mobile big data poses a great challenge, and recent studies on this issue can be found in [110]–[112].

2) Fog/Edge Computing: In order to reduce the network delay coming from the backhaul network, the concept of fog computing [2] (as shown in Fig. 9) is proposed to bring the computing and storage capability closer to the mobile devices, near the edge of the network. In other words, devices located at the edge of the Internet, such as routers, switches, base stations, access points, etc., will be equipped with computing and storage resources. In fact, fog computing extends cloud computing schemes from the core of the network to the edge

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

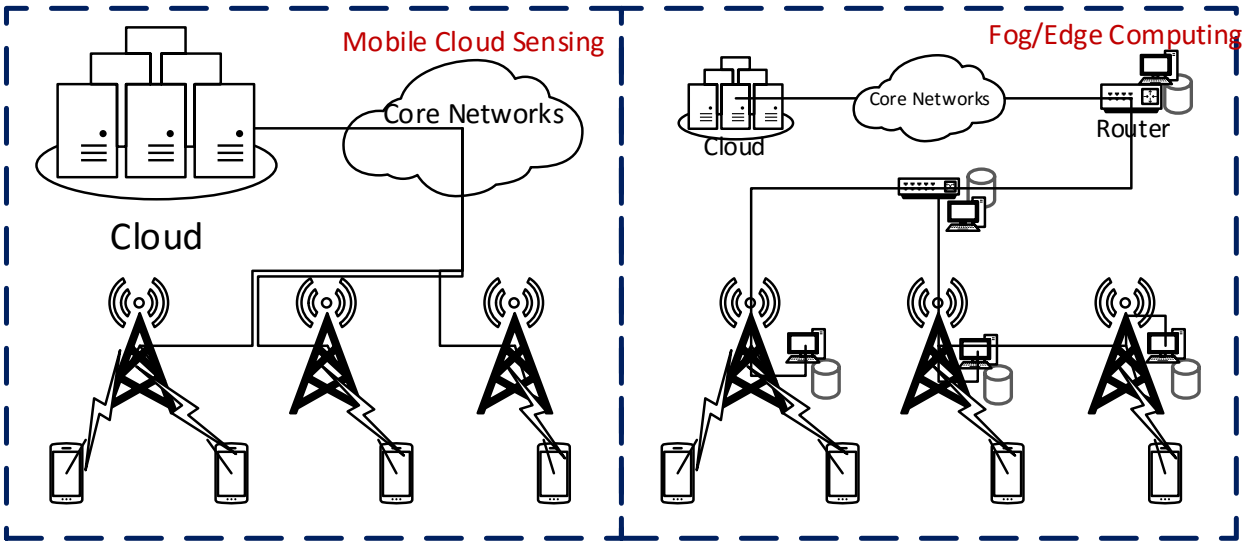


Fig. 9. Computing Paradigms to Support Mobile Big Data

of the network.

In the context of mobile big data, the fog computing paradigm can deal with data acquisition, aggregation and preprocessing, and even data mining, without suffering from the high latency as in mobile cloud computing. However, the computing and storage resources of a single network device at the edge of the network may not have sufficient capability to handle the mobile big data tasks, such that cooperation among edge devices with limited individual computing capability is of great interest. The concept of cloudlet is to form a cloud-like computing paradigm based on multiple edge devices with computing resources in physical proximity, in order to both reduce the latency and provide powerful computing resources via mobile device cooperation [109], [113]. Accordingly, efficient heterogeneous computing resource management in a hierarchical network poses research challenges and provides great research opportunities. In particular, the interaction and coordination control among the edge devices leads to many intriguing research problems.

Although the paradigm of fog computing can reduce the latency to the core of the Internet, the bandwidth and connectivity limitation in the current structure of wireless access networks (especially in the widely used cellular networks) is still present.

3) Software Defined Networking (SDN): The difficulty of reducing the latency of the core network largely comes from the distributed nature of the computer network. In fact, network functionalities could be divided into three hierarchical planes: data, control and management [114]. At each network device, the data plane forwards the data packets and the control plane implements the protocols in order to populate the forwarding table for the data plane. The management plane is to monitor and configure the control plane.

In recent years, the idea of software defined networking (SDN) is proposed to cope with the control issue of computer networks, by centralizing the control plane of individual network devices to an external entity (Fig. 10(a)). In other words, the data plane is decoupled from the control plane and remotely controlled [114]. With SDN, the forward decisions are based on network flows (defined as a sequence of packets between a source and a destination) rather than the destination of packets. Atop the centralization of the control plane, network applications and services in the management plane, such as routing, firewall, load balancing, status monitoring and so on, are implemented based on programmable interfaces provided by the centralized SDN controller.

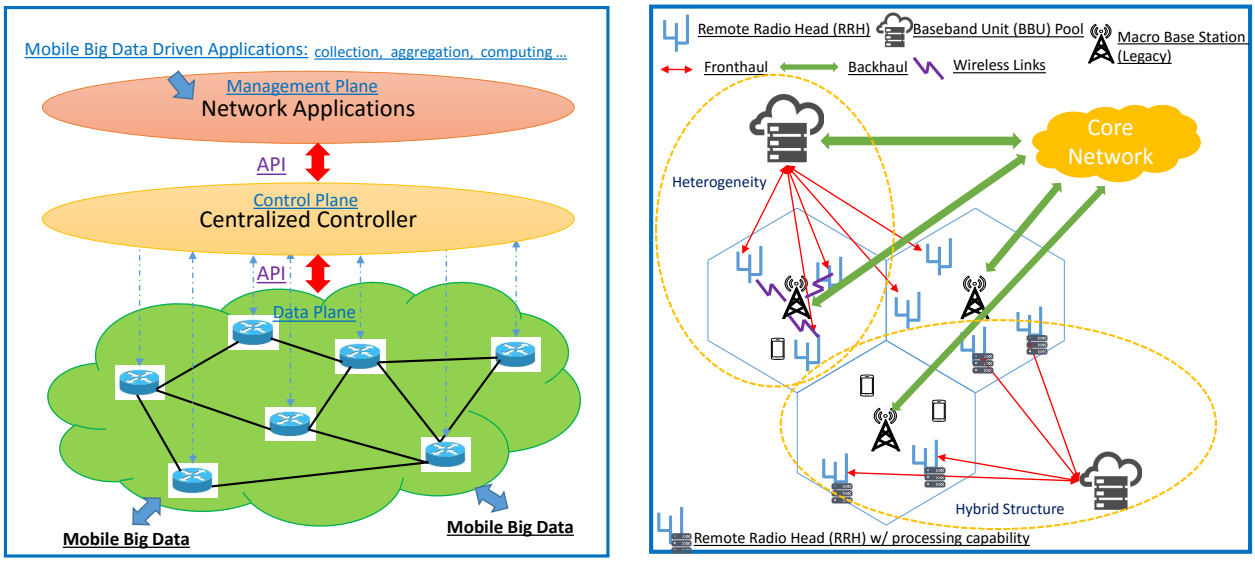
B. Communication and Networking Infrastructure

In the context of mobile big data, network performance is a key factor that connects mobile terminals and the cloud computing platform. With the development of SDN, network latency may be improved with specific network applications deployed on the centralized control plane. However, there are still challenges in the context of big data applications [115], [116]. For example, the (mobile) big data applications (computing and processing) postulate more rapid and frequent flow table updates, in order to fulfill the needs of bulk data transfer, data aggregation/partition, and so on, in the context of distributed big data computing and storage. This leads to various design and implementation issues in SDN.

1) Cloud Radio Access Networks (C-RAN): The unprecedented volume of mobile big data traffic will bring great challenges to current radio access networks (RANs), namely cellular networks in our context, which are generally used in mobile data collection and transmission. The current RAN bandwidth and capacity are not able to fulfill the demand of mobile big data applications. Therefore, the paradigm of RAN needs to be revolutionized.

In the traditional RAN, base stations (BSs) with limited number of antennas can only serve a fixed coverage, which leads to the underutilization of network resources over both space and time. In the evolution of RAN, small cells are preferred to increase the spatial spectrum reuse. However, the interference management and coordination in the hierarchical

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal



(a) Software Defined Networking (SDN)

(b) Cloud Radio Access Networks (C-RAN)

Fig. 10. Communications and Networking Paradigms to Support Mobile Big Data

cell structure post great challenges. In addition, the computing resource in the traditional BSs may not be able to fulfill the demands of dynamic resource management.

The concept of C-RAN [65], [117] is proposed to centralize the computation-intensive functions (baseband processing and resource management) into the backend cloud connected to BSs via high-capacity connections, which can be wired (like optical fiber) or wireless. Meanwhile, the only function that remains in BSs is the RF-level wireless accessing and possibly some simple symbol processing. Therefore, the radio access networks are essentially divided into two parts, remote radio head (RRH) for RF accessing and baseband unit (BBU) pool for processing, as shown in Fig. 10(b). The transition from distribution to centralization for baseband processing brings great advantages [107] on load balancing, interference management, multi-cell coordination, etc. In addition, the network device parameters could be reconfigured online, which will provide great flexibility and full utilization of the network resources for high-capacity and reliable radio access services.

With the decoupling between wireless access functions and computing functions, multi-cell joint dynamic resource allocation could be facilitated spatially and temporally in C-RAN, according to the learned user mobility patterns. Furthermore, optimal collaborative radio processing could also be enabled in the centralized computing paradigm of C-RAN, as mobile users could usually be served by multiple small cells. Nevertheless, the design of dynamic resource allocation and collaboration of radio processing to support the real-time high-data-rate applications of mobile big data are facing many open problems.

On the other hand, the computing cloud is able to learn and predict the behaviors of users with the availability of joint spatial and temporal mobile data from the users. The learned knowledge will in turn provide guidance to adjust network structures and reconfigure device parameters, such that the network performance and quality of service can be optimized under the architecture of C-RAN. However, it is challenging to identify and extract useful features from massive mobile big data, as well as to discover the underlying relationship linking mobile user behaviors and network performance.

With the learned knowledge on user behavior, one could cache popular contents in the BSs of macro cells, small cells or even some user devices, which could potentially improve the quality of experience by reducing the content downloading delay, as the content cached at the edge of the network is closer to users. In the literature, caching can be applied not only at the application layer, but also at the network layer [118] or even at the data link layer [119]. However, determination of what to cache is challenging in cache-assisted communication and networking. Generally, the Zipf distribution [120], [121] is assumed to characterize the popularity of contents in most existing results. Although it is well studied that the content popularity follows the Zipf distribution as a whole, it is not accurate to assume that the popularity of contents still follows the Zipf distribution locally in a small region. Therefore, the content popularity as well as the user demand profiles should be further learned from the mobile data that local users generated.

Indeed, the centralization of baseband processing functionality poses great stresses and challenges on connections bridging the front-end RRHs and the back-end BBUs, due to the network capacity constraints, which will limit the performance of the overall system. To deal with the capacity constraints of such connections, Bi *et al.* in [65] re-considered the scheme of computing resource allocation and proposed a hybrid computing structure to cope with this limited capacity problem mentioned above. Specifically, some computing tasks are proposed to remain at BSs to reduce the transmission burden to/from the cloud. Peng *et al.* in [122] proposed to utilize some high-power BSs as a fronthaul for control

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

signal broadcasting, which not only reduces the transmission burden to/from the cloud but also mitigates the heterogeneous coordination problem between the C-RAN and the traditional cellular networks. Indeed, the tradeoff between the centralized and the distributed computing of radio access networks is still an open problem, together with the heterogeneous coordination between C-RAN and traditional cellular networks.

VII. Computing

Mobile big data analytics demand high performance computing to accommodate the "5V" features and other distinct characteristics, in order to facilitate potential applications and services. In fact, sequential computing at a single machine cannot fulfill such computational demand over the tremendous amount of data, and a single machine may not even be able to hold the entire dataset in its memory, especially as the Moore's law is fading nowadays. Therefore, parallel computing over multiple nodes is of great importance in the era of (mobile) big data.

In this section, we provide an overview on existing computing solutions, which may be adopted for mobile big data analytics to better match the special characteristics of mobile big data (e.g., real-time). A complete (mobile) big data computing solution consists of two main components: the computing platform (hardware) and the computing architecture (software). In fact, a system of large-scale distributed commodity machines working parallelly in a distributed manner is a generally preferred computing platform in terms of flexibility and capital cost. Such a large-scale distributed computing platform has gained popularity along with the maturing software development of large-scale distributed computing [123], [124], which is deployed atop the hardware. In particular, the software is developed to efficiently utilize the large-scale distributed hardware resources, whose system architecture generally consists of three layers, namely the data injection layer, the data analytic layer, and the data storage layer. Each of them focuses on a distinct functionality in such a large-scale distributed computing system. In the rest of this section, we will first introduce the large-scale distributed computing platform (hardware), and then present the key properties required for large-scale distributed computing (software), as well as the other details of its system architecture.

A. Large-Scale Computing Platform (Hardware)

In the era of (mobile) big data, the fact that the complexity of big data processing exceeds the computing capability of a single node leads to massive parallel computing, which consists of multiple processors. In general, the very nature of a massive parallel computing system is determined by its two main components (as shown Fig. 11), namely the participating processors and the network architecture connecting those processors. Heterogeneous computing arises when processors of different types, e.g., central processing units (CPUs) and graphic processing units (GPUs), are involved in parallel computing. The networking among the participating processors, on the other hand, naturally plays a critical role in a system containing a massive number of processors in that it fundamentally determines the coupling level of nodes and in turn the capability of the system. In the sequel, we will discuss the computing platform for big data processing from these two perspectives.

A.1. Heterogeneous Computing

The legacy principle of a parallel system design is to interconnect the multiple similar processors into one system [125], which is termed as homogeneous computing. On the contrary, heterogeneous computing aims to exploit the strengths of different types of processors with intelligent load balancing to improve the computational performance [126].

In heterogeneous computing, the specialized accelerators, e.g., GPUs or reconfigurable logics (FPGA), are integrated into machines with CPUs to accelerate the computing. The accelerators are connected to CPUs via an external bus, from which the heterogeneity arises. The commodity GPUs are the most commonly-used accelerators, which are originally designed for the game industry in image and graphic computing and later extended to support parallel data computing (termed as the general purpose GPU (GPGPU)). The power of GPUs lies in their thousands of cores, which could provide up to a few tera float operations per seconds (TFLOPS). In fact, the price/performance ratio is one of the most important advantages of the GPU-accelerated heterogeneous computing, compared with a cluster of nodes with similar computing capability without GPUs. For example, the NVIDIA 1080 graphic card could provide about 9 TFLOPS, which costs only a few hundred US dollars.

The CUDA package from NVIDIA provides the application programming interfaces (APIs) to utilize any CUDA-enabled GPUs for general computing [127]. The parallelism of CUDA-enabled GPU could be hierarchically divided into two levels: the GPU has multiprocessors (MPs); each MP has multiple stream processors (SPs). The SPs share a fast but small memory (typical 16 KB), based on which threads run in the SPs could synchronize their states. A global memory, larger but relatively slower (8 GB in a NVIDIA 1080 graphic card), is shared across the MPs. A collection of threads, termed as a block, is scheduled to feed a MP, in which the SPs are the ones that actually carry out the computations [128]. The parallelism in GPUs is accomplished by independent computations performed in the SPs. However, the performance gain of GPUs could be bottlenecked by the communication bandwidth between the GPUs and the CPUs [129]. Once the data or model size exceeds a GPU's global memory capacity, the frequent communications and data exchanges between the CPU and the GPUs would largely reduce the computing gain. Therefore, another level of parallelism and scalability has been developed to mitigate this problem, in which data or models are divided and alloted to a cluster of GPUs or a cluster of machines accelerated by GPUs [130]–[132].

A.2. Computing Systems

The actual operation of computing systems with a massive number of nodes heavily depends on the network connecting all these nodes. In the one extreme, multiple nodes in proximity are tightly interconnected with a dedicated local network, such that the system with multiple nodes could be regarded as a single virtual machine. This is the case with high

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

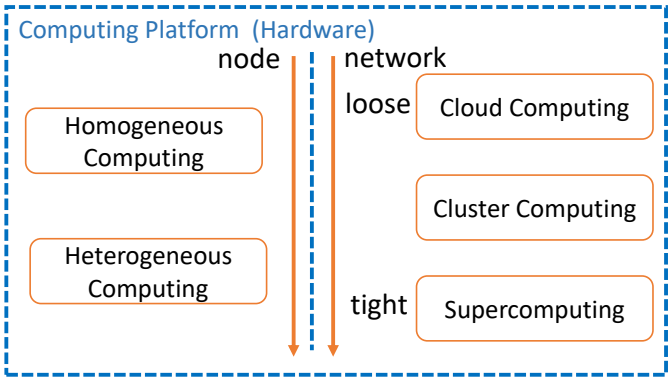


Fig. 11. Large-Scale Computing Platform

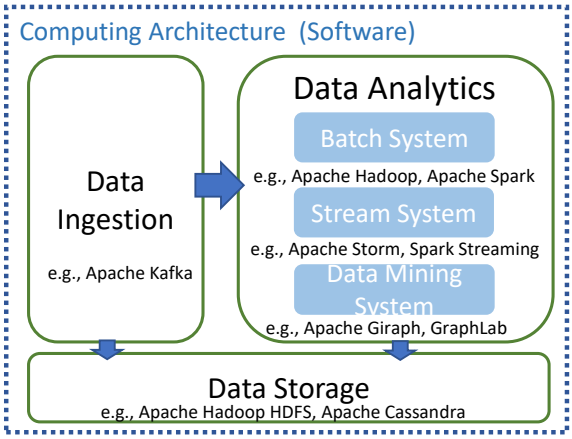


Fig. 12. Large Scale Distributed Computing Architecture

performance computing (a.k.a., supercomputing). Supercomputing could date back to the 1960s, which was introduced by Seymour Cray [133]. The supercomputer usually consists of tens of thousands of processors in proximity, which is regarded as a single virtual machine. At the beginning, the architecture with customized vector processors and (distributed) shared memory is generally utilized in supercomputers, where the memories are accessed in a uniform addressing space. However, the architecture consisting of commodity processors and distributed memories has gained popularity in 2000s, where data exchange and sharing are fulfilled via message passing. In both cases, however, the interconnection of nodes in a supercomputer plays a critical role in the resultant computing performance. The interconnection in a supercomputer is generally accomplished by a customized dedicated high-speed local network, such as InfiniBand [134] (e.g., IBM Roadrunner) and Torus [133] (e.g., Cray XE6). The supercomputer originally designed for large volume data processing and computationally intensive simulations is also suitable for (mobile) big data analytics. However, the supercomputer with large-scale multiple processors tightly connected by a customized interconnection technology requires enormous capital investment, which is generally not available to big data researchers. In addition, the centralized administration of supercomputers leads to extra costs on the system operation and maintenance (e.g., cooling system).

In the other extreme, multiple nodes, possibly geographically distributed, are connected via, e.g., the Internet, such that a large computing task could be divided into multiple small tasks executed independently at each node. A well known example of this is the so-termed cloud computing. Cloud computing consists of multiple nodes that are often geographical distributed with computing and storage capability. These nodes are might be loosely connected by the Internet. The cloud computing is aimed to enable ubiquitous, convenient, and on-demand public computing services [135] hosted by vendors, such as Amazon Web Service (AWS), Microsoft Azure, and Google App Engine. Three service models, namely infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS), provide different levels of pay-as-your-go price models in terms of the amount of consumed computing resources. The business models of cloud computing are enabled by a virtualization layer between the physical hardware and the operating system at each node. The virtualization layer [136] is an abstraction of the underlying hardware including processors, memory, storage, and networks, which acts as a finer-grained computing resource unit, fulfilling the flexible on-demand computing requests. That is, a physical node could host several logical machines and the usage of the latter determines the actual cost of a particular computing task. However, to host the data at a public infrastructure may lead to privacy concerns, especially for the privacy-sensitive mobile big data. Details of big data analytics via cloud computing can be found in [137].

Lying in the middle ground between the two aforementioned extremes, cluster computing is an interconnected multi-node system without a customized high-speed local networks. Generally, a cluster may refer to a computing system that consists of multiple (tens to hundreds) commodity processors interconnected by general Ethernet networks. In the literature, this is termed as Beowulf cluster [133]. In addition, the cluster with multiple nodes working together closely could also be regarded as one single machine, which runs an non-proprietary OS. In fact, cluster computing could be sometimes regarded as a small-scale supercomputer [138], [139]. On the one hand, the utilization of commodity networking technology could largely reduce the capital investment, due to the economies-of-scale effect on general commercial hardwares. On the other hand, the limited communication capability of cluster computing leads to a bottleneck of performance, which greatly reduces the applicability for those which require large-scale frequent data exchange and sharing. Nevertheless, cluster computing is capable of dealing with task-independent applications. That is, a large task could be divided into small tasks that could be executed independently without frequent data exchange. Most of (mobile) big data problems could be formulated in this manner such that cluster computing can be applied.

B. Large-Scale Distributed Computing Architecture (Software)

The large-scale computing software layer not only provides an easy-to-use interface for users to implement their data analytic algorithms as a single- or multi-thread program,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

but also facilitates utilization of the distributed computing resources provided by the hardware layer.

B.1. Key Properties & Architecture

In a large-scale computing, the parallelization of the programming or computing paradigm could be roughly categorized into two types, namely algorithmic parallelization and data parallelization [140]. The algorithmic parallelization essentially divides the algorithm into multiple tasks, which could be simultaneously processed at multiple computing nodes. The challenging issue of algorithmic parallelization is the communication among the processes at different nodes. The message passing interface (MPI) [141] is a typical standard on communications in distributed parallel computing among multiple nodes. In fact, the algorithmic parallelization is suitable for complex algorithms, which is usually applied to both streaming computational systems and data mining systems. On the other hand, data parallelization is more intuitive in the era of big data, as the data could be partitioned into multiple groups and each node in the system could process each data group independently while focusing on a single computing task. Hence, data parallelization is always adopted in the batch computational system. Nevertheless, all program models for scalable and parallel computing should satisfy some key properties as detailed below.

- **Scalability.** The scalability allows a large-scale computing system to utilize all distributed computing resources efficiently. In fact, a new computing architecture needs to be carefully designed in term of scalability to accommodate a tremendous amount of data by minimizing the communication requests from a large number of commodity machines (which could be scaled to tens of thousands of machines).
- **Fault Tolerance and Recovery.** Due to the massive number of nodes at the large-scale data center (tens of thousands of nodes), faults are inevitable. Faults may result from network congestions, disk errors, or scheduled offline maintenance. In fact, unreachable nodes are usually also recognized as faults. The computing system should be able to tolerate such faults, by automatically rescheduling jobs at the faulted nodes to standby nodes and restoring the intermediate results from the failing nodes. In addition, a robust system design should facilitate easy recovery from human faults, as those are also inevitable.
- **Robustness to Stragglers.** Stragglers refer to the slow nodes in a system, which hold the entire computing process from proceeding. The sluggishness of computation at a node may be due to the competition of computing resources (e.g., CPU, memory, local disk) among multiple tasks. The program model should be able to deal with stragglers and reassign their jobs to alternative nodes.
- **Data Locality.** The major bottlenecks of a distributed computing system are the I/O of disks and the network bandwidth. Due to the massive data volume, the computing paradigm would transfer the computing task to nodes with the data rather than moving data across the network.

In the literature, there are many projects related to large-scale distributed computing. However, none of them can provide a one-size-fits-all solution for all (mobile) big data analytics problems. A real-time (mobile) big data analytics system should have three main components (shown in Fig. 12), namely the *data ingestion layer*, the *data analytic layer* and the *data storage layer* [142].

- **Data Ingestion Layer.** The data injection layer is aimed to absorb and queue the data from multiple sources, and to provide asynchronous communications, with a certain degree of fault tolerance, between the data source and data processing units. A typical open source data injection layer is the Apache Kafka [143], which manages data in a message-passing manner based on the publish and subscribe scheme.
- **Data Analytic Layer.** Inside the data analytic layer, there are also three components: the batch system, the stream system, and the data mining system [142]. Actually, the computation on the entire raw dataset is laid upon the the batch computational system, due to its high throughput. However, batch computation is usually applied to the entire data set, resulting in a high latency. The output of the batch computational system is often outdated when it is ready. Hence, a stream computation system is developed to process recent data streams that the batch computational system cannot handle. The stream computational system could be of high throughput as well as low latency while dealing with much smaller data sizes. In addition, large-scale data mining is aimed to accommodate data mining or machine learning algorithms efficiently in the big data computing system.

  A high-level architecture, Lambda Architecture [123], further details the data analytic layer. Besides the batch system and the stream system, a serve system is separated from the batch system to store outputs of batch computing (a.k.a, batch views in [123]). The results corresponding to the user query (a function on the whole set of data) will be computed based on both the batch views and the real-time views that are respective outputs of the batch computational system and stream computational system. In fact, the batch computational system will reproduce the batch views by re-computing the entire available dataset periodically rather than simply integrating the stream views with its outdated batch views. This will enhance the tolerance against the stream computational system faults or even the batch computational system faults. However, the robustness of such a re-computation scheme is at the cost of performance loss, as the batch system always needs to process the entire dataset.
- **Data Storage Layer.** To enhance the reliability of a big data system, the raw data stored in the big data system is designed to be eternal and immutable, such that any results could be recovered by re-computations on the immutable raw data. Hence, the data storage layer should provide a reliable storage service for the constantly growing master dataset. The scalability of a storage system is very much desired, since holding the massive volume of data as a single machine is neither

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

practical nor reliable. In addition, the high-performance parallel reading should be supported for scalable batch computing in the data analytic layer. Distributed file systems are good candidates for the storage of raw data, for they provide fault tolerance and scalability [123]. The most commonly-used distributed file system is the Hadoop distributed file system (HDFS), which is a critical component of Apache Hadoop [144].

Next, we will discuss some widely adopted systems for the data analytic systems with more details.

B.2. Data Analytic Layer Computing Systems

1) Batch Computing Systems

The concept of MapReduce, proposed by Google in 2004 [145], is a popular batch computing paradigm based on data parallelization. Although it has gained popularity in various applications, MapReduce has some inherent drawbacks such as high latency for iterative algorithms. To address such limitations, AMPLab of the University of California, Berkeley, proposed a new programming model, namely the resillient distributed dataset (RDD) in 2010 [146].

- **MapReduce.** The MapReduce programming model is proposed to handle the massive data volume by a cluster of thousands of nodes in a parallel manner [145]. The open-source implementation of MapReduce is Apache Hadoop [144], which is a complete solution of distributed cluster computing based on MapReduce. MapReduce consists of two key tasks, map and reduce. The input data is first partitioned into multiple groups, which are then processed in parallel by map functions of independent nodes or workers (mappers). Intermediate results produced by mappers are stored locally (usually on disks) at each node. After all map tasks are completed, the intermediate results will be shuffled and fed to reduce tasks (reducers), which further process the intermediate results to final results as desired. The final results are saved at a global storage. A master node schedules computational jobs among workers and monitors their respective status (idle, in-progress or completed).
  As MapReduce is designed to operate on multiple machines, the scalability issue is well resolved in MapReduce. In terms of fault tolerance, the master node in MapReduce will periodically ping every worker. Unreachable workers are regarded as failing nodes. The failed map tasks will be re-executed at other nodes, as intermediate results are stored at the local storage. The failed reduce tasks, however, may not need to be re-executed, as the final results are saved at the global storage. In the original implementation of MapRduce, a global file system (GFS) divides each file into multiple equi-length blocks, and keeps multiple copies (typically 3 copies) at the local storage. The master node generally assigns a map task to a node that contains a replica [147] to achieve data locality. When a MapReduce procedure is near completion, the master node will schedule redundant executions for tasks that are still in progress, in order to mitigate the negative effects caused by stragglers.
- **Resilient Distributed Dataset (RDD).** Due to the limitation of disk I/O, the in-memory computing with minimum data exchanges between the memory and the disk will significantly expedite the computation, especially when the in-memory working dataset will be reused in an iterative analysis algorithm. In MapReduce, each iteration would be expressed as a MapReduce job, which is inefficient [148] as the whole procedure involves lots of disk I/O operations. However, most mobile big data analytic algorithms will involve a great amount of iterative steps. In [149], Zaharia *et al.* proposed a distributed memory abstraction, termed as resilient distributed dataset (RDD), to support in-memory computation, while still maintaining MapReduce's attractive properties, such as fault tolerance, locality-aware scheduling, and scalability. The open-source implementation of RDD is Apache Spark [146].
  An RDD is defined as a read-only partitioned collection of data, which could only be generated through deterministic operations (termed as transformation in Apache Spark) on either datasets in storage or other existing RDDs. After RDDs are defined, the computation will not be executed until an *action* is called, where the *action* is defined as an operation that returns values or exports data to a storage system. The partition of RDD is designed to support the property of scalability. The partition order could be controlled by users based on a key associated with each data record. In addition, the intermediate result caching could also be controlled and specified by users.
  The reason why the RDD generation is restricted to deterministic operations on existing RDDs or datasets in stable storage, is to lower the overhead of the fault tolerance support. In general, two methods are employed to support fault tolerance: checkpointing (replicating the data) and data update logging (recording the differences). In fact, the replication of data across the network is expensive due to its massive volume; so is data update logging if many updates are made. Instead of recording the data itself, the RDD system will remember a series of deterministic operations (a.k.a., lineage [149]) that generate the current RDD. When an RDD partition is lost due to faults, the system will regenerate the current RDD based on the recorded lineage. In fact, the success of RDD relies on the sufficiency of the memory at each node, in comparison with MapReduce. When memories at a system are not sufficient, the performance gain of RDD could be marginal [150].

2) Stream Computing Systems

As mentioned in Section II, the distinct "real time" feature of mobile big data not only refers to the high velocity of the data stream, but also emphasizes the required responding speed of mobile big data based applications. In the literature, streaming computing is aimed to deal with such situations in large clusters (capable of cloud or cluster computing), but still maintaining the scalability, fault-tolerance, and locality properties.

- **Two Types of Stream Computational Systems.** Unlike batch processing on the entire available dataset,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

the stream computing system is aimed to catch the recent data that the batch system cannot handle. The streaming computational system could be categorized into two types: one-at-a-time and micro-batched, respectively [123]. The one-at-a-time streaming computational system incrementally processes each data record in a predefined data stream one after another following some continuous models. The scalability of the one-at-a-time system lies in that the processing on each data record could be performed in parallel across multiple nodes in the system, i.e., algorithmic parallelization. One-at-a-time stream processing projects include Apache S4 [151], Apache Storm [152], and TimeStream [153]. In one-at-a-time systems, every data record will possibly trigger a new update on the real-time view by flowing through a predefined computing logic network, whose topology could be characterized by a directed acyclic graph (DAG). The DAG computational topology adopted in the one-at-a-time system will also simplify the design and implementation of algorithmic parallelization.

On the other hand, the micro-batched system first processes a small batch of data records at a time following a discretized model and then updates the real-time view by integrating the output of the current micro batch processing and the past real-time view. An example micro-batched system is D-Stream (Spark Streaming) [154], which utilizes the RDD computing paradigm in each micro batch process. Evidently, the one-at-a-time system has much lower latency, as it does not have to wait for a preset time interval. In addition, it takes more time to process a set of data records than just a single record, while the processing time is relatively much smaller than that of the micro-batched processing. Nevertheless, the micro-batched system has a higher throughput and is more robust than the one-at-a-time system, in which the fault and straggler could be recovered easily and rapidly. A survey and benchmark of existing streaming computational systems could be found in [155], [156].

- **Fault-Tolerant Methods.** Two fault-tolerant methods are adopted in the one-at-a-time systems: the hardware replication and the upstream backup [154]. The hardware replication introduces redundancy to the system, by simultaneously keeping two sets of nodes. The upstream backup is to replay the entire computation on the backup data from the very beginning (e.g., Apache S4) or from the intermediate result (e.g., Apache Storm). The re-computation on the data from the very beginning in Apache S4 leads to very high fault recovery latency. The re-computation in Apache Storm is not general and is only applicable to some special operators [153]. In TimeStream, the system keeps monitoring the dependencies of the output and the state at each node in the computational network, and the lost output of the failed node is re-computed from its immediate saved states. However, the fault-tolerant method in batch processing, such as lineage in RDD, could be extended to each small batch computing task, as the real-time views are updated at the batch level and small batch processing tasks are independent of each other, which will enhance the fault tolerance.

3) Data Mining Systems

In mobile big data, the data from mobile users may not only be treated independently as in batch or stream processing, but also be jointly investigated in terms of their relative relationship or dependency. For example, the location of mobile users is studied [44] along with their social relationship. In general, the relationship among the data could be characterized by graphs (directed or undirected). In fact, the dependencies in data limit the applicability of MapReduce, since MapReduce requires strict data parallelism. The MapReduce-based iterative processing is significantly inefficient, due to the lack of direct iterative mechanisms in MapReduce. In addition, many machine learning or data mining algorithms update their parameters iteratively, e.g., training of deep belief networks. Therefore, efficient distributed computing in such scenarios is of a significant demand in mobile big data processing. Pregel [157] and GraphLab [158] are two typical graph-parallel abstractions in the literature. GraphX [159] implements both Pregel and GraphLab abstractions with a graph extension of RDD in Spark, termed as resilient distributed graph (RDG).

- **Pregel and Its Derivative.** The pregel computing system is initially aimed to efficiently deploy large-scale graph computing in a cluster of commodity computers efficiently, which is built by Google [157] and its open-source version is maintained in Apache Giraph [160].

  Similar to the MapReduce model, the graph is partitioned in terms of the vertices of the input graph and assigned to each node of the cluster. The parallelization of Pregel is bulk synchronous parallel (BSP). That is, Pregel synchronously carries out simultaneous computation on all vertices of the graph at each iteration of a graph algorithm, where each iteration is termed as a "superstep" in Pregel. The interaction of vertices in a graph algorithm is fulfilled by messages passed by other vertices of the graph in the previous iteration. In addition, messages pushed by vertices in one superstep could only be processed in the next superstep. In other words, the computing on the entire graph is synchronized at the granularity of supersteps and messages shuffle across the entire cluster before a new superstep begins. Such global synchronism makes the graph algorithm implementation easier and free of deadlocks or data races, but leads to high latency in general. The message passing model in Pregel improves the efficiency compared to MapReduce, since it does not need to update the state of the entire graph at each iteration as the chained MapReduce does. The fault-tolerance in Pregel is ensured by checkpointing, where nodes periodically store the states of each vertex at a global storage and the entire graph will be re-computed from the latest check point when one or more nodes fail. The message staleness issue will arise when a message could not be seen and processed immediately but needs to wait until the next superstep. This will compromise the system performance due to node blocking for strict global synchronism on the entire graph. Giraph++ proposed in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2714189, IEEE Internet of Things Journal

[161] aims to mitigate such an issue by allowing message exchanges among internal vertices within a subgraph to be immediately seen and processed. Hence, the computing of a graph moves from a vertex-centric manner to a graph-centric manner. GiraphUC proposed in [162] further reduces the communications and node blocking overhead due to global synchronization across the entire graph, by splitting a global superstep into multiple logical supersteps. The logical superstep is locally synchronized, which could reduce a large portion of communication blockings and stragglers in the system due to global synchronization. Such a programming model is termed as barrierless asynchronous parallel (BAP). In fact, Pregel and Giraph++ could both be regarded as some BAP special cases where a global superstep is divided into multiple logical supersteps. However, such hybrid models may lead to high coordination complexity in the system and compromise the ease of use for parallel graph algorithm implementation.

- **GraphLab.** The GraphLab was originally developed for parallel graph computing, machine learning and data mining at a single machine with multiple cores [163]. It was later adapted to cluster computing in [158], [164]. GraphLab is also a vertex-centric abstraction. Unlike Pregel, GraphLab completely eliminates the synchronism requirement in message passing. Instead, Graphlab applies the shared memory to achieve asynchronism, which will in turn accelerate the convergence of a machine learning algorithm. The shared memory allows the vertex to read and write the state of each vertex following a pull model. The dynamic schedule of GraphLab, in which not all parameters are updated in the universal and uniform frequency could further reduce the number of iterations required for convergence. This is achieved by decoupling the computation schedule and the data movement in the system. However, asynchronism may lead to data inconsistency due to data races. Hence, data lock is demanded in GraphLab to ensure correctness. In GraphLab, different distributed lock models could be employed to solve this issue. However, distributed locking is resource-expensive and may lead to performance degradation. In addition, the termination of a computation task needs to be checked by a distributed consensus algorithm, due to the lack of global synchronization in GraphLab.

The significant recent advancement in big data computing makes mobile big data processing possible with a cluster of commodity computers. However, the computing system design for a specific mobile big data task varies in terms of the data volume, data velocity, and specific requirements of the task. In general, all the subsystems described previously might be combined based on their advantages and the needs of specific mobile big data applications.

VIII. Data Security

The mobile big data containing the information of time, location, and even activities of users is always privacy-sensitive. Furthermore, such privacy-sensitive mobile big data is not just confined at user terminals or remote servers but also at any device with computing and storage capabilities in the network, such as the routers, switches, access points, etc., due to the distributed nature of fog/edge computing. Therefore, how to protect the data and preserve the user privacy [165], while processing services by collecting mobile big data, poses great challenges.

1) Authentication: User authentication and data encryption are two key factors to ensure user privacy. The biometric identification [166], including iris, voice, fingerprint, face, etc., is reliable and convenient for user authentication. However, the biometric information itself is privacy-sensitive, which requires careful encryption and protection [167], [168], especially when cloud storage is used for such biometric data. Protection of the biometric information and efficient design of encryption algorithms remain open problems. This becomes more challenging if computing is required to be transparent to privacy protection. In [168], Chun *et al.* proposed a privacy-preserving biometric authentication protocol to outsource the computationally heavy authentication process to the cloud without compromising user privacy, in which the biometric data is fully encrypted before being sent to the cloud and is never decrypted during the authentication process in the cloud.

2) Access control: The distributed computing and storage resources in the cloud serve as the infrastructure to support various mobile big data applications, where one node with computing and storage resources could be accessed by multiple agents. Then, the data life cycle in the distributed storage should be carefully designed to prevent data leakage. In addition, valuable user data is always a big attraction of attacks and hacks [169]. How to identify and detect unauthorized accesses will be a challenging but interesting issue. In [170], Stolfo *et al.* attempted to identify and detect abnormal access behaviors in the context of data theft, based on user behavior profiling which models when and how a user accesses his/her data in the cloud. In addition, bogus information will be generated and returned by the cloud in order to confuse the adversary, once an unauthorized access is detected.

Furthermore, the access control for data sharing is another challenging problem in the context of mobile big data. The attribute-based encryption (ABE) [171] allows data owners to specify a set of users (termed as data access policy) who can access the data with an authorized attribute. When the data owner wants to change the data access policy, one traditionally needs to re-encrypt the data according the new policy by moving the data from the cloud to the local site and then move the newly encrypted data back to the cloud. However, the back and forth data transfer is almost impossible under mobile settings due to the tremendous data volume and limited transport capacity. In [172], Yang *et al.* proposed a method to update the access policy directly in the cloud without retrieving and decrypting the data.

IX. Conclusions

In this survey paper, we introduced the unique features of mobile big data, its sources and various applications. Rooted on these, in-depth discussions of existing research

results, along with research opportunities and challenges, are laid out in terms of user modeling, computing infrastructure, communication and networking architecture, data security and privacy, and knowledge discovery. All these are essential in bringing the envisioned future of mobile big data applications into reality. We hope that, with this comprehensive survey, more researchers and developers will be inspired to devote efforts to this emerging research field with great potentials.

REFERENCES

[1] X. Cheng, L. Fang, X. Hong, and L. Yang, "Exploiting mobile big data: Sources, features, and applications," *IEEE Network*, vol. 31, no. 1, pp. 72–79, Jan. 2017.

[2] Y. Cao, P. Hou, D. Brown, J. Wang, and S. Chen, "Distributed analytics and edge intelligence: Pervasive health monitoring at the era of fog computing," in *Proceedings of the 2015 Workshop on Mobile Big Data*, Hangzhou, China, Jun. 22–25, 2015, pp. 43–48.

[3] R. LiKamWa, Y. Liu, N. D. Lane, and L. Zhong, "Moodscope: Building a mood sensor from smartphone usage patterns," in *Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, Taipei, Taiwan, Jun. 25–28, 2013, pp. 389–402.

[4] Q. Han, S. Liang, and H. Zhang, "Mobile cloud sensing, big data, and 5G networks make an intelligent and smart world," *IEEE Network*, vol. 29, no. 2, pp. 40–45, Mar. 2015.

[5] D. Laney, "3D data management: Controlling data volume, velocity and variety," *META Group Research Note*, vol. 6, p. 70, Feb. 2001.

[6] X. Dong and D. Srivastava, "Big data integration," in *Proceedings of the 29th IEEE International Conference on Data Engineering (ICDE)*, Brisbane, Australia, Apr. 8–12, 2013, pp. 1245–1248.

[7] J. Gantz and D. Reinsel, "Extracting value from chaos," *IDC iView*, no. 1142, pp. 9–10, 2011.

[8] M. Ficek and L. Kencl, "Inter-call mobility model: a spatio-temporal refinement of call data records using a Gaussian mixture model," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, Orlando, FL, Mar. 25–30, 2012, pp. 469–477.

[9] A. Ladd, K. Bekris, G. Marceau, A. Rudys, D. Wallach, and L. Kavraki, "Using wireless Ethernet for localization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, Sep. 30–Oct. 4, 2002, pp. 402–408.

[10] B. Ferris, D. Fox, and N. D. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 7, Hyderabad, India, Jan. 6–12, 2007, pp. 2480–2485.

[11] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, "Efficient, generalized indoor WiFi GraphSLAM," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9–13, 2011, pp. 1038–1043.

[12] M. Balakrishnan, I. Mohomed, and V. Ramasubramanian, "Where's that phone?: geolocating IP addresses on 3G networks," in *Proceedings of the 9th ACM SIGCOMM conference on Internet Measurement Conference*, Chicago, Illinois, Nov. 4–6, 2009, pp. 294–300.

[13] A. Metwally and M. Paduano, "Estimating the number of users behind IP addresses for combating abusive traffic," in *Proceedings of the 17th ACM International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, Aug. 21–24, 2011, pp. 249–257.

[14] L. T. Le, T. Eliassi-Rad, F. Provost, and L. Moores, "Hyperlocal: Inferring location of IP addresses in real-time bid requests for mobile Ads," in *Proceedings of the 6th ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, Orlando, FL, Nov. 5, 2013, pp. 24–33.

[15] H. Hu and D.-L. Lee, "Semantic location modeling for location navigation in mobile environment," in *Proceeding of IEEE International Conference on Mobile Data Management (MDM)*, Berkeley, CA, Jan. 19–22, 2004, pp. 52–61.

[16] B. Shaw, J. Shea, S. Sinha, and A. Hogue, "Learning to rank for spatiotemporal search," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, Rome, Italy, Feb. 4–8, 2013, pp. 717–726.

[17] J. Goncalves, S. Hosio, D. Ferreira, and V. Kostakos, "Game of words: Tagging places through crowdsourcing on public displays," in *Proceedings of the 2014 Conference on Designing Interactive Systems*, Vancouver, Canada, Jun. 7–11, 2014, pp. 705–714.

[18] N. Stojanovic, L. Stojanovic, Y. Xu, and B. Stajic, "Mobile CEP in real-time big data processing: Challenges and opportunities," in *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, Mumbai, India, May 26–29, 2014, pp. 256–265.

[19] M. Mardani, G. Mateos, and G. B. Giannakis, "Subspace learning and imputation for streaming big data matrices and tensors," *IEEE Transactions on Signal Processing*, vol. 63, no. 10, pp. 2663–2677, May 2015.

[20] A. Tajer, V. V. Veeravalli, and H. V. Poor, "Outlying sequence detection in large data sets: A data-driven approach," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 44–56, Sep. 2014.

[21] D. Z. Yazti and S. Krishnaswamy, "Mobile big data analytics: Research, practice, and opportunities," in *Proceedings of the 15th IEEE International Conference on Mobile Data Management (MDM)*, Brisbane, QLD, Jul. 14–18, 2014, pp. 1–2.

[22] A. Cavoukian, "Introduction to privacy by design," 2016. [Online]. Available: https://www.ipc.on.ca/english/Privacy/Introduction-to-PbD/

[23] J. K. Laurila, D. Gatica-Perez, I. Aad, J. Blom, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen, "The mobile data challenge: Big data for mobile computing research," in *Proceedings of Nokia Workshop on Mobile Data Challenge in conjunction with International Conference on Pervasive Computing*, Newcastle, UK, Jun. 18–20, 2012.

[24] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila, "Towards rich mobile phone datasets: Lausanne data collection campaign," in *Proceedings of ACM International Conference on Pervasive Services*, Berlin, German, Jul. 13–16, 2010, pp. 1–7.

[25] I. Aad and V. Niemi, "NRC data collection and the privacy by design principles," *Phone Sense*, pp. 41–45, Nov. 2010.

[26] M. Musolesi, "Big mobile data mining: Good or evil?" *IEEE Internet Computing*, vol. 18, no. 1, pp. 78–81, Jan. 2014.

[27] M. L. Damiani, E. Bertino, and C. Silvestri, "The PROBE framework for the personalized cloaking of private locations," *ACM Transactions on Data Privacy*, vol. 3, no. 2, pp. 123–148, Aug. 2010.

[28] M. Damiani, C. Silvestri, and E. Bertino, "Fine-grained cloaking of sensitive positions in location-sharing applications," *IEEE Pervasive Computing*, vol. 10, no. 4, pp. 64–72, Apr. 2011.

[29] Y. Park and E. Lee, "A new generation method of a user profile for information filtering on the internet," in *Proceedings of the 12th International Conference on Information Networking*, Tokyo, Japan, Jan. 21-23, 1998, pp. 261–264.

[30] R. J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *Proceedings of the 15th ACM Conference on Digital Libraries*, San Antonio, TX, 2000, pp. 195–204.

[31] M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill & webert: Identifying interesting web sites," in *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, OR, Aug. 4–8, 1996, pp. 54–61.

[32] W. Kim, L. Kerschberg, and A. Scime, "Learning for automatic personalization in a semantic taxonomy-based meta-search agent," *Electronic Commerce Research and Applications*, vol. 1, no. 2, pp. 150–173, Summer 2002.

[33] C. C. Tossell, P. Kortum, C. W. Shepard, A. Rahmati, and L. Zhong, "Getting real: a naturalistic methodology for using smartphones to collect mediated communications," *Human-Computer Interaction*, vol. 2012, no. 10, pp. 1–10, Apr. 2012.

[34] E. Nathan and A. Pentland, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, Mar. 2006.

[35] D. Wagner, A. Rice, and A. Beresford, "Device analyzer: Understanding smartphone usage," in *Proceedings of the 10th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, Tokyo, Japan, Dec. 2–4, 2013, pp. 195–208.

[36] D. T. Wagner, A. Rice, and A. R. Beresford, "Device analyzer: Large-scale mobile data collection," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 4, pp. 53–56, Mar. 2014.

[37] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, Mar. 2008.

[38] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, Feb. 2010.

[39] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Scientific Reports*, vol. 3, Mar. 2013.

[40] J. Krumm and E. Horvitz, "Predestination: Inferring destinations from partial trajectories," in *Proceedings of the 8th International Conference*

on Ubiquitous Computing, Orange County, CA, Sep. 17–21, 2006, pp. 243–260.

[41] N. Eagle and A. Pentland, "Eigenbehaviors: identifying structure in routine," *Behavioral Ecology and Sociobiology*, vol. 63, no. 7, pp. 1057–1066, May 2009.

[42] A. Sadilek and J. Krumm, "Far out: Predicting long-term human mobility." in *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, Toronto, Canada, Jul. 22–26, 2012.

[43] B. D. Ziebart, A. L. Maas, A. K. Dey, and J. A. Bagnell, "Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, Seoul, Korea, Sep. 21–24, 2008, pp. 322–331.

[44] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, Aug. 21–24, 2011, pp. 1082–1090.

[45] P. A. Grabowicz, J. J. Ramasco, B. Gonalves, and V. M. Eguiluz, "Entangling mobility and interactions in social media," *PLoS ONE*, vol. 9, no. 3, p. e92196, Mar. 2014.

[46] M. De Domenico, A. Lima, and M. Musolesi, "Interdependence and predictability of human mobility and social interactions," *Pervasive and Mobile Computing*, vol. 9, no. 6, pp. 798–807, Dec. 2013.

[47] J. L. Toole, C. Herrera-Yaqüe, C. M. Schneider, and M. C. González, "Coupling human mobility and social ties," *Journal of The Royal Society Interface*, vol. 12, no. 105, pp. 1–9, Feb. 2015.

[48] J. Zhuang, T. Mei, S. C. Hoi, Y.-Q. Xu, and S. Li, "When recommendation meets mobile: Contextual and personalized recommendation on the go," in *Proceedings of the 13th International Conference on Ubiquitous Computing*, Beijing, China, Sep. 17–21, 2011, pp. 153–162.

[49] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, and J. Tian, "Mining mobile user preferences for personalized context-aware recommendation," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 4, pp. 58:1–58:27, Dec. 2014.

[50] R. K. Wong, V. W. Chu, and T. Hao, "Online role mining for context-aware mobile service recommendation," *Personal Ubiquitous Computing*, vol. 18, no. 5, pp. 1029–1046, Jun. 2014.

[51] R. Kumar, M. Mahdian, B. Pang, A. Tomkins, and S. Vassilvitskii, "Driven by food: Modeling geographic choice," in *Proceedings of the 8th ACM International Conference on Web Search and Data Mining (WSDM)*, Shanghai, China, Jan. 1–Feb. 6, 2015, pp. 213–222.

[52] F. Sposaro and G. Tyson, "iFall: An android application for fall monitoring and response," in *Proceedings of the Annual International Conference of Engineering in Medicine and Biology Society (EMBC)*, Minneapolis, MN, Sep. 3–6, 2009, pp. 6119–6122.

[53] W. Wu, S. Dasgupta, E. E. Ramirez, C. Peterson, and J. G. Norman, "Classification accuracies of physical activities using smartphone motion sensors," *Journal of Medical Internet Research*, vol. 14, no. 5, p. e130, Oct. 2012.

[54] V. Sharma, K. Mankodiya, F. De La Torre, A. Zhang, N. Ryan, T. G. Ton, R. Gandhi, and S. Jain, "SPARK: Personalized parkinson disease interventions through synergy between a smartphone and a smartwatch," in *Design, User Experience, and Usability. User Experience Design for Everyday Life Applications and Services*, A. Marcus, Ed. Grete, Greece: Springer, 2014, pp. 103–114.

[55] A. Moore, "Why 2016 could be a watershed year for emotional intelligence–in machines," 2016. [Online]. Available: http://blogs.scientificamerican.com/guest-blog/why-2016-could-be-a-watershed-year-for-emotional-intelligence-in-machines/

[56] J. M. Girard, J. F. Cohn, and F. De la Torre, "Estimating smile intensity: A better way," *Pattern Recognition Letters*, vol. 66, pp. 13–21, Nov. 2015.

[57] S. Grauwin, S. Sobolevsky, S. Moritz, I. Gdor, and C. Ratti, "Towards a comparative science of cities: Using mobile traffic records in New York, London, and Hong Kong," in *Computational Approaches for Urban Environments*, ser. Geotechnologies and the Environment, M. Helbich, J. Jokar Arsanjani, and M. Leitner, Eds. Springer International Publishing, Nov. 2015, vol. 13, pp. 363–387.

[58] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and POIs," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, China, Aug. 12–16, 2012, pp. 186–194.

[59] M. Berlingerio, F. Calabrese, G. Lorenzo, R. Nair, F. Pinelli, and M. L. Sbodio, "AllAboard: A system for exploring urban mobility and optimizing public transport using cellphone data," in *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases*, Prague, Czech Republic, Sep. 23–27, 2013, pp. 663–666.

[60] H. Dong, M. Wu, X. Ding, L. Chu, L. Jia, Y. Qin, and X. Zhou, "Traffic zone division based on big data from mobile phone base stations," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 278–291, Sep. 2015.

[61] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: Driving directions based on taxi trajectories," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, San Jose, CA, Nov. 2–5, 2010, pp. 99–108.

[62] F. Antonelli, M. Azzi, M. Balduini, P. Ciuccarelli, E. D. Valle, and R. Larcher, "City sensing: Visualizing mobile and social data about a city scale event," in *Proceedings of International Working Conference on Advanced Visual Interfaces*, Como, Italy, May 27–30, 2014, pp. 337–338.

[63] B. Moumni, V. Frias-Martinez, and E. Frias-Martinez, "Characterizing social response to urban earthquakes using cellphone network data: The 2012 Oaxaca earthquake," in *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, Zurich, Switzerland, Sep. 8–12, 2013, pp. 1199–1208.

[64] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "Dnn-based prediction model for spatio-temporal data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Burlingame, California, Oct. 31 – Nov. 3, 2016, pp. 92:1–92:4.

[65] S. Bi, R. Zhang, Z. Ding, and S. Cui, "Wireless communications in the era of big data," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 190–199, Aug. 2015.

[66] S. Hoteit, S. Secci, Z. He, C. Ziemlicki, Z. Smoreda, C. Ratti, and G. Pujolle, "Content consumption cartography of the Paris urban region using cellular probe data," in *Proceedings of the 1st Workshop on Urban Networking*, Nice, France, Dec. 10–13, 2012, pp. 43–48.

[67] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 6.

[68] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug 2013.

[69] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[70] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012.

[71] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[72] I. G. Y. Bengio and A. Courville, "Deep learning," 2016, book in preparation for MIT Press. [Online]. Available: http://www.deeplearningbook.org

[73] M. A. Alsheikh, D. Niyato, S. Lin, H. p. Tan, and Z. Han, "Mobile big data analytics using deep learning and apache spark," *IEEE Network*, vol. 30, no. 3, pp. 22–29, May 2016.

[74] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 2672–2680.

[75] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, Jan. 2014.

[76] M. Agosti and M. Melucci, "Information retrieval on the web," in *Proceedings of the 3rd European Summer-School on Lectures on Information Retrieval*, Varenna, Italy, Sep. 11–15, 2000, pp. 242–285.

[77] C. Buckley, A. Singhal, and M. Mitra, "Using query zoning and correlation within SMART," in *Proceedings of the 5th Text Retrieval Conference*, Gaithersburg, Maryland, Nov. 1996, pp. 105–118.

[78] G. Salton, "The smart retrieval system," in *Experiments in Automatic Document Processing*. Prentice Hall, 1971.

[79] L. Shastri, "Why semantic networks," in *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. Morgan Kaufmann, 1991.

[80] S. Thomas, "Http essentials: protocols for secure, scaleable web sites," in *Scalable Web Sites*. John Wiley, New York, Mar. 2001.

[81] Y. H. Wu, Y. C. Chen, and L. P. Chen, "Enabling personalized recommendation on the web based on user interests and behaviors," in *Proceedings of the 11th International Workshop on Research Issues in Data Engineering*, Heidelberg, Germany, 2001, pp. 17–24.

[82] "Open directory project." [Online]. Available: https://www.dmoz.org/

[83] R. Cooley, B. Mobasher, and J. Srivastava, "Web mining: information and pattern discovery on the World Wide Web," in *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence*, Newport Beach, CA, Nov. 3–8, 1997, pp. 558–567.

[84] B. Mobasher, "Web usage mining," *Web data mining: Exploring hyperlinks, contents and usage data*, vol. 12, pp. 1216–1220, 2005.

[85] J. Srivastava, R. Cooley, M. Deshpande, and P. Tan, "Web usage mining: discovery and applications of usage patterns from web data," *ACM SIGKDD Explorations Newsletter*, vol. 1, no. 2, pp. 12–23, Jan. 2000.

[86] L. Ungar and D. Foster, "Clustering methods for collaborative filtering," in *Proceedings of the AAAI Workshop on Recommendation Systems*, Madison, WI, Jul. 26–27, 1998.

[87] A. Kohrs, A. Kohrs, B. Merialdo, and B. Merialdo, "Clustering for collaborative filtering applications," in *Computational Intelligence for Modelling, Control & Automation. Intelligent Image Processing, Data Analysis & Information Retrieval*, M. Mohammadian, Ed. IOS Press, 1999, pp. 199–204.

[88] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, Jan. 1977.

[89] X. Fu, J. Budzik, and K. J. Hammond, "Mining navigation history for recommendation," New Orleans, LA, Jan. 9–12, 2000, pp. 106–112.

[90] W. Lin, S. A. Alvarez, and C. Ruiz, "Efficient adaptive-support association rule mining for recommender systems," *Data Mining and Knowledge Discovery*, vol. 6, no. 1, pp. 83–105, Jan. 2002.

[91] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Effective personalization based on association rule discovery from web usage data," in *Proceedings of the 3rd International Workshop on Web Information and Data Management*, Atlanta, Georgia, Nov. 5–10, 2001, pp. 9–15.

[92] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommender algorithms for e-commerce," in *Proceedings of the 2nd ACM Conference on Electronic Commerce*, Minneapolis, MN, Oct. 17–20, 2000, pp. 158–167.

[93] B. Suryavanshi, N. Shiri, and S. Mudur, "Improving the effectiveness of model based recommender systems for highly sparse and noisy web usage data," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Compiegne, France, Sep. 19–22, 2005, pp. 618–621.

[94] Y. Zhou, X. Jin, and B. Mobasher, "A recommendation model based on latent principle factors in web navigation data," in *Proceedings of the 3rd International Workshop on Web Dynamics in conjunction with the 13th International World Wide Web Conference*, New York City, NY, May 18, 2004, pp. 52–61.

[95] I. Cadez, P. Smyth, E. Ip, and H. Mannila, "Predictive profiles for transaction data using finite mixture models," Information and Computer Science Department, University of California, Irvine, Irvine, CA, Tech. Rep. 01-67, Dec. 2001.

[96] M. Kaixiang, T. Ben, Z. Erheng, and Y. Qiang, "Report of Task 3: Your Phone Understands You," in *Proceedings of the Workshop on Nokia Mobile Data Challenge*, Newcastle, UK, Jun. 18-19, 2012.

[97] W. M. C. Nadeem S, "Demographic prediction of mobile user from phone usage," in *Proceedings of the Workshop on Nokia Mobile Data Challenge*, Newcastle, UK, Jun. 18-19, 2012, pp. 16–21.

[98] J. Ying, Y. Chang, C. Huang, and V. Tseng, "Demographic prediction based on users mobile behaviors," in *Proceedings of the Workshop on Nokia Mobile Data Challenge*, Newcastle, UK, Jun. 18-19, 2012.

[99] J. Wang, C. Zeng, C. He, L. Hong, L. Zhou, R. Wong, and J. Tian, "Context-aware role mining for mobile service recommendation," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, Trento, Italy, Mar. 26–30 2012, pp. 173–178.

[100] L. Qian, B. Wu, R. Zhang, W. Zhang, and M. Luo, "Characterization of 3G data-plane traffic and application towards centralized control and management for software defined networking," in *Proceedings of IEEE International Congress on Big Data*, Santa Clara, CA, Jun. 27–Jul. 2, 2013, pp. 278–285.

[101] L. Wang, K. Hu, T. Ku, X. Yan, and L. Wang, "Mining frequent trajectory pattern based on vague space partition," *Knowledge-Based Systems*, vol. 50, pp. 100–111, Sep. 2013.

[102] X. Wu, K. N. Brown, and C. J. Sreenan, "Analysis of smartphone user mobility traces for opportunistic data collection in wireless sensor networks," *Pervasive and Mobile Computing*, vol. 9, no. 6, pp. 881–891, Dec. 2013.

[103] Z. Sun and X. J. Ban, "Vehicle classification using GPS data," *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 102–117, Dec. 2013.

[104] D. J. Yang W, Cheng H, "A location-aware recommender system for mobile shopping environments," *Expert Systems with Applications*, vol. 34, no. 1, pp. 437–445, Jan. 2008.

[105] W. Hsu, G. Jacobsen, Y. Jin, and A. Skudlark, "Using social media data to understand mobile customer experience and behavior," in *Proceedings of the 22nd European Regional Conference of the International Telecommunications Society*, Budapest, Hungary, Sep. 18–21, 2011.

[106] C. Herrera-Yagüe and P. J. Zufiria, "Prediction of telephone user attributes based on network neighborhood information," in *Proceedings of the 8th International Conference on Machine Learning and Data Mining in Pattern Recognition*, Berlin, Germany, Jul. 13–20, 2012, pp. 645–659.

[107] Y. Cai, F. Yu, and S. Bu, "Cloud computing meets mobile wireless communications in next generation cellular networks," *IEEE Network*, vol. 28, no. 6, pp. 54–59, Nov. 2014.

[108] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 369–392, First 2014.

[109] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[110] A. Vulimiri, P. B. Godfrey, R. Mittal, J. Sherry, S. Ratnasamy, and S. Shenker, "Low latency via redundancy," in *Proceedings of the 9th ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, Santa Barbara, CA, Dec. 9–12, 2013, pp. 283–294.

[111] B. Bangerter, S. Talwar, R. Arefi, and K. Stewart, "Networks and devices for the 5G era," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 90–96, Feb. 2014.

[112] A. Vulimiri, O. Michel, P. B. Godfrey, and S. Shenker, "More is less: Reducing latency via redundancy," in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, Redmond, WA, Oct. 29–30, 2012, pp. 13–18.

[113] M. Quwaider and Y. Jararweh, "Cloudlet-based efficient data collection in wireless body area networks," *Simulation Modelling Practice and Theory*, vol. 50, pp. 57–71, Jan. 2015.

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

[115] G. Wang, T. Ng, and A. Shaikh, "Programming your network at run-time for big data applications," in *Proceedings of the 1st ACM workshop on Hot topics in Software Defined Networks (HotSDN)*, Helsinki, Finland, Aug. 13–17, 2012, pp. 103–108.

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

[117] S.-H. Park, O. Simeone, O. Sahin, and S. Shamai, "Joint precoding and multivariate backhaul compression for the downlink of cloud radio access networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5646–5658, Nov. 2013.

[118] H. Ahlehagh and S. Dey, "Video-aware scheduling and caching in the radio access network," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1444–1462, Oct. 2014.

[119] A. Liu and V. K. N. Lau, "Cache-enabled opportunistic cooperative MIMO for video streaming in wireless systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 390–402, Jan. 2014.

[120] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, San Diego, CA, Oct. 24–26, 2007, pp. 1–14.

[121] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[122] M. Peng, K. Zhang, J. Jiang, J. Wang, and W. Wang, "Energy-efficient resource assignment and power allocation in heterogeneous cloud radio access networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5275–5287, Nov. 2015.

[123] N. Marz and J. Warren, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*, 1st ed. Greenwich, CT, USA: Manning Publications Co., 2015.

[124] D. A. Reed and J. Dongarra, "Exascale computing and big data," *Communication of ACM*, vol. 58, no. 7, pp. 56–68, Jun. 2015.

[125] Y. Gao and P. Zhang, "A survey of homogeneous and heterogeneous system architectures in high performance computing," in *Proceeding*

of IEEE International Conference on Smart Cloud (SmartCloud), Nov 2016, pp. 170–175.

[126] S. Mittal and J. S. Vetter, "A survey of cpu-gpu heterogeneous computing techniques," *ACM Computing Survey*, vol. 47, no. 4, Jul. 2015.

[127] M. Harris, "Many-core gpu computing with nvidia cuda," in *Proceedings of the 22nd Annual International Conference on Supercomputing*, Island of Kos, Greece, Jun. 7–12, 2008, pp. 1–1.

[128] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09, Montreal, Quebec, Canada, Jun. 14–17, 2009, pp. 873–880.

[129] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. aurelio Ranzato, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng, "Large scale distributed deep networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1223–1231.

[130] Z. Fan, F. Qiu, A. Kaufman, and S. Yoakum-Stover, "Gpu cluster for high performance computing," in *Proceedings of the 2004 ACM/IEEE Conference on Supercomputing*, Pittsburgh, PA, Nov. 2004, pp. 47–47.

[131] V. V. Kindratenko, J. J. Enos, G. Shi, M. T. Showerman, G. W. Arnold, J. E. Stone, J. C. Phillips, and W. m. Hwu, "Gpu clusters for high-performance computing," in *Proceedings of IEEE International Conference on Cluster Computing*, New Orleans, LA, USA, Aug. 31 – Sep. 4, 2009, pp. 1–8.

[132] A. Coats, B. Huval, T. Wang, D. J. Wu, and A. Y. Ng, "Deep learning with COTS HPC systems," in *Proceedings of the 30 th International Conference on Machine Learning*, Atlanta, Georgia, Jun. 16–21, 2013.

[133] S. Limet, W. W. Smari, and L. Spalazzi, "High-performance computing: to boldly go where no human has gone before," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 13, pp. 3145–3165, 2015.

[134] G. F. Pfister, "An introduction to the infiniband architecture," in *High Performance Mass Storage and Parallel I/O*, J. Fagerberg, D. C. Mowery, and R. R. Nelson, Eds., 2001, ch. 42, pp. 617–632.

[135] P. M. Mell and T. Grance, "Sp 800-145. the nist definition of cloud computing," Gaithersburg, MD, United States, Tech. Rep., 2011.

[136] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[137] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of big data on cloud computing: Review and open research issues," *Information Systems*, vol. 47, pp. 98 – 115, Jan. 2015.

[138] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Proceedings of Grid Computing Environments Workshop*, Austin, TX, Nov. 16, 2008, pp. 1–10.

[139] N. Sadashiv and S. M. D. Kumar, "Cluster, grid and cloud computing: A detailed comparison," in *Proceeding of the 6th International Conference on Computer Science Education (ICCSE)*, Singapore, Aug. 3–5, 2011, pp. 477–482.

[140] H. Mohamed and S. Marchand-Maillet, "Distributed media indexing based on MPI and MapReduce," *Multimedia Tools and Applications*, vol. 69, no. 2, pp. 513–537, Nov. 2014.

[141] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall, *Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 97–104.

[142] R. Ranjan, "Streaming big data processing in datacenter clouds," *IEEE Cloud Computing*, vol. 1, no. 1, pp. 78–83, May 2014.

[143] "Apache Kafka," 2016. [Online]. Available: http://kafka.apache.org/

[144] "Apache Hadoop," 2016. [Online]. Available: http://hadoop.apache.org/

[145] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation*, vol. 6, San Francisco, CA, Dec. 5, 2004, pp. 137–149.

[146] "Apache Spark," 2016. [Online]. Available: http://spark.apache.org/

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

[148] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing*, Boston, MA, Jun. 22–25, 2010, pp. 1–7.

[149] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, San Jose, CA, Apr. 25–27, 2012, pp. 1–15.

[150] L. Gu and H. Li, "Memory or time: Performance evaluation for iterative operation on Hadoop and Spark," in *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications & IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC)*, Zhangjiajie, China, Nov. 13–15, 2013, pp. 721–727.

[151] L. Neumeyer, B. Robbins, A. Nair, and A. Kesari, "S4: Distributed stream computing platform," in *Proceedings of IEEE International Conference on Data Mining Workshops*, Sydney, Australia, Dec. 13, 2010, pp. 170–177.

[152] "Apache Storm," 2016. [Online]. Available: http://storm.apache.org/

[153] Z. Qian, Y. He, C. Su, Z. Wu, H. Zhu, T. Zhang, L. Zhou, Y. Yu, and Z. Zhang, "Timestream: Reliable stream computation in the cloud," in *Proceedings of the 8th ACM European Conference on Computer Systems*, Prague, Czech Republic, Apr. 14–17, 2013, pp. 1–14.

[154] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Stoica, "Discretized streams: Fault-tolerant streaming computation at scale," in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, Farminton, Pennsylvania, Nov. 3–6, 2013, pp. 423–438.

[155] X. Liu, N. Iftikhar, and X. Xie, "Survey of real-time processing systems for big data," in *Proceedings of the 18th International Database Engineering & Applications Symposium*, Porto, Portugal, Jul. 7–9, 2014, pp. 356–361.

[156] S. Qian, G. Wu, J. Huang, and T. Das, "Benchmarking modern distributed streaming platforms," in *Proceedings of IEEE International Conference on Industrial Technology (ICIT)*, Taipei, Taiwan, Mar. 14–17, 2016, pp. 592–598.

[157] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: A system for large-scale graph processing," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, Indianapolis, Indiana, USA, Jun. 6–10, 2010, pp. 135–146.

[158] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein, "Distributed graphlab: A framework for machine learning and data mining in the cloud," *Proceedings of VLDB Endowment*, vol. 5, no. 8, pp. 716–727, Apr. 2012.

[159] R. S. Xin, J. E. Gonzalez, M. J. Franklin, and I. Stoica, "Graphx: A resilient distributed graph system on spark," in *Proceedings of First International Workshop on Graph Data Management Experiences and Systems*, New York City, NY, Jun. 23, 2013, pp. 2:1–2:6.

[160] "Apache Giraph," 2016. [Online]. Available: http://giraph.apache.org/

[161] Y. Tian, A. Balmin, S. A. Corsten, S. Tatikonda, and J. McPherson, "From "think like a vertex" to "think like a graph"," *Proceedings of VLDB Endowment*, vol. 7, no. 3, pp. 193–204, Nov. 2013.

[162] M. Han and K. Daudjee, "Giraph unchained: Barrierless asynchronous parallel execution in pregel-like graph processing systems," *Proceedings of VLDB Endowment*, vol. 8, no. 9, pp. 950–961, May 2015.

[163] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein, "Graphlab: A new framework for parallel machine learning," *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, Jul. 8–11, 2010.

[164] J. E. Gonzalez, Y. Low, H. Gu, D. Bickson, and C. Guestrin, "Powergraph: Distributed graph-parallel computation on natural graphs," in *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*, Hollywood, CA, Oct. 8–10, 2012, pp. 17–30.

[165] C. Tankard, "Big data security," *Network Security*, vol. 2012, no. 7, pp. 5–8, Jun. 2012.

[166] S. Sharma and V. Balasubramanian, "A biometric based authentication and encryption framework for sensor health data in cloud," in *Proceedings of International Conference on Information Technology and Multimedia (ICIMU)*, Putrajaya, Malaysia, Nov. 18–20, 2014, pp. 49–54.

[167] Q. A. Kester, L. Nana, A. C. Pascu, S. Gire, J. M. Eghan, and N. N. Quaynor, "Feature based encryption technique for securing forensic biometric image data using AES and visual cryptography," in *Proceedings of the 2nd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*, Madrid, Spain, Nov. 18–20, 2014, pp. 199–204.

[168] H. Chun, Y. Elmehdwi, F. Li, P. Bhattacharya, and W. Jiang, "Outsourceable two-party privacy-preserving biometric authentication," in *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, Kyoto, Japan, Jun. 4–6, 2014, pp. 401–412.

[169] Z. Tan, U. T. Nagar, X. He, P. Nanda, R. P. Liu, S. Wang, and J. Hu, "Enhancing big data security with collaborative intrusion detection," *IEEE Cloud Computing*, vol. 1, no. 3, pp. 27–33, Sep. 2014.

[170] S. J. Stolfo, M. B. Salem, and A. D. Keromytis, "Fog computing: Mitigating insider data theft attacks in the cloud," in *Proceedings of IEEE Symposium on Security and Privacy Workshops (SPW)*, San Francisco, CA, May 24–25, 2012, pp. 125–128.

[171] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of IEEE Symposium on Security and Privacy*, Berkeley, CA, May 20–23, 2007, pp. 321–334.

[172] K. Yang, X. Jia, and K. Ren, "Secure and verifiable policy update outsourcing for big data access control in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3461–3470, Dec. 2015.

Xiang Cheng (S'05-M'10-SM'13) received the Ph.D. degree from Heriot-Watt University and the University of Edinburgh, Edinburgh, U.K., in 2009, where he received the Postgraduate Research Thesis Prize. He is currently an Associate Professor at Peking University. His general research interests are in areas of channel modeling and mobile communications for 5G and VANET, subject on which he has published more than 160 journal and conference papers, 3 books and 6 patents.

Dr. Cheng was the recipient of the IEEE Asia Pacic (AP) Outstanding Young Researcher Award in 2015, the co-recipient for the 2016 IEEE JSAC Best Paper Award: Leonard G. Abraham Prize, the NSFC Outstanding Young Investigator Award, the Second-Rank Award in Natural Science, Ministry of Education in China, and received the Best Paper Awards at IEEE ITST12, ICCC13, ITSC14, ICC16, and ICNC17. He has served as Symposium Leading-Chair, Co-Chair, and a Member of the Technical Program Committee for several international conferences. He is now an Associate Editor for IEEE Transactions on Intelligent Transportation Systems.

Luoyang Fang (S'12) received his B.S. degree in Department of Electronics and Information Engineering from Huazhong University of Science and Technology, Wuhan, China, in 2011, and is now pursuing his Ph.D. degree in Department of Electrical and Computer Engineering, Colorado State University. His research interests include big data, mobile data, location privacy, data mining, distributed storage system and information-centric networking.

Liuqing Yang (S'02-M'04-SM'06-F'15) received the Ph.D. degree from the University of Minnesota, Minneapolis, MN, USA, in 2004. She is currently a Professor with the Department of Electrical and Computer Engineering at Colorado State University. Her main research interests include communications and signal processing. Dr. Yang has been actively serving in the technical community, including the organization of many IEEE international conferences, and on the editorial boards of a number of journals, including the IEEE Transactions on Communications, the IEEE Transactions on Wireless Communications, the IEEE Transactions on Intelligent Transportation Systems, and the IEEE Transactions on Signal Processing. She received the Office of Naval Research Young Investigator Program Award in 2007, the National Science Foundation Career Award in 2009, the IEEE GLOBECOM Outstanding Service Award in 2010 the George T. Abell Outstanding Mid-Career Faculty Award at CSU in 2012, and Best Paper Awards at IEEE ICUWB'06, ICCC'13, ITSC'14, GLOBECOM'14, ICC'16 and WCSP'16.

Shuguang Cui (S99-M05-SM12-F14) received his Ph.D in Electrical Engineering from Stanford University, California, USA, in 2005. Afterwards, he has been working as assistant, associate, and full professor in Electrical and Computer Engineering at the Univ. of Arizona and Texas A&M University. He is currently a Childs Family Endowed Professor in Electrical and Computer Engineering at the Univ. of California-Davis. His current research interests focus on data driven large-scale information analysis and system design, including large-scale distributed estimation and detection, information theoretical approaches for large data set analysis, complex cyber-physical system design, and cognitive network optimization. He was selected as the Thomson Reuters Highly Cited Researcher and listed in the Worlds Most Influential Scientific Minds by ScienceWatch in 2014. He was the recipient of the IEEE Signal Processing Society 2012 Best Paper Award. He has served as the general co-chair and TPC co-chairs for many IEEE conferences. He has also been serving as the area editor for IEEE Signal Processing Magazine, and associate editors for IEEE Transactions on Big Data, IEEE Transactions on Signal Processing, IEEE JSAC Series on Green Communications and Networking, and IEEE Transactions on Wireless Communications. He was the elected member for IEEE Signal Processing Society SPCOM Technical Committee (2009 2014) and the elected Chair for IEEE ComSoc Wireless Technical Committee (2017 2018). He is a member of the Steering Committee for both IEEE Transactions on Big Data and IEEE Transactions on Cognitive Communications and Networking. He is also a member of the IEEE ComSoc Emerging Technology Committee. He was elected as an IEEE Fellow in 2013 and an IEEE ComSoc Distinguished Lecturer in 2014.