A Perspective on the Challenges and Opportunities for Privacy-Aware Big Transportation Data

Godwin Badu-Marfo · Bilal Farooq · Zachary Patterson

Received: 10 July 2018 / Revised: 7 November 2018 / Accepted: 9 November 2018 / Published online: 4 April 2019
© Springer Nature Singapore Pte Ltd. 2019

Abstract
In recent years, and especially since the development of the smartphone, enormous amounts of data relevant for transportation have become available. These data hold out the potential to redefine how transportation system (i.e., design, planning and operations) is done. While researchers in both academia and industry are making advances in using this data to transportation system ends (e.g., information inference from collected data), little attention has been paid to four larger scale challenges that will need to be overcome if the potential for Big Transportation Data is to be harnessed for transportation decision-making purposes. This paper aims to provide awareness of these large-scale challenges and provides insight into how we believe these challenges are likely to be met.

Keywords Big Data · Human travel mobility · Cloud computing · Privacy awareness · Scalability

Introduction
Transportation system (i.e., design, planning and operations) has been a quantitative discipline highly dependent upon data at least since the birth of modern travel demand modeling in the 1950s. Until recently, data collection has been done through dedicated, often self-reported surveys (e.g., household surveys, on-board surveys, etc.), and through various methodologies and technologies concentrated on vehicle flow counts (e.g., loop detectors). Recently, a combination of devices and technologies have dramatically increased the number of potential sources, as well as the amount of data that can be collected with urban transportation system applications, what we refer to as Big Transportation Data. Examples of this data include Bluetooth and CCTV traffic counts (Barcelo et al. 2010; Cathey and Dailey 2005), pedestrian counts with Wi-Fi (Poucin et al. 2018; Danalet et al. 2014; Farooq et al. 2015), activity detection with social media location data (Yazdizadeh et al. 2018), dedicated travel survey smartphone applications (Patterson and Fitzsimmons 2017) and smartphone data aggregators (StreetLight 2018).

The potential for this data in transportation systems have not been overlooked, with many researchers in academia and the public and private (Lv et al. 2015; Dong et al. 2015; Zheng et al. 2016; Chen et al. 2016) sectors investigating ways in which to use it in their processes. Until now, the academic literature has been primarily preoccupied with two aspects of Big Data in transportation. First there has been research on how to go about collecting relevant data with these new technologies (e.g., Patterson and Fitzsimmons 2017; Leduc 2008; Shi and Abdel-Aty 2015). Second, there has been research focusing on methods (statistical, machine learning, etc.) using collected data and inferring transportation relevant information from it (e.g., mode, trip purpose, etc.) (Yazdizadeh et al. 2018; Nitsche et al. 2014; Zhang et al. 2014).

While the successful collection of data, and inference of information relevant to transportation system presents many challenges to the routine incorporation of Big Transportation Data in design, planning and operations, little attention has been paid to the impending challenge of actually being able to store, manage and process all the data on large and operational scale, not to mention the challenge of protecting...
privacy of the people providing the data. We divide these large-scale implementation challenges into four dimensions. The tautological fact that there is a large quantity of Big Data presents challenges in storing it. Second, the need to compute algorithms on large-scale data presents a challenge in processing. Third, Big Data comes in many different formats, making the ability to take advantage of data collected from different sources challenging. Fourth is the challenge of protecting personal privacy.

While the quantity of data and diversity of formats are primarily technical challenges, personal privacy is a political as well as technical challenge. The political nature of the challenge was recently evidenced by the controversy around Facebook and Cambridge Analytica (Solon 2018) and public reaction to it. The issue of privacy and Big Data is multifaceted. Most obviously, much Big Data is sufficiently detailed (e.g., geographically and temporally precise GPS data) that it could reasonably be used to identify individuals. A less obvious challenge to privacy is the ability to combine information about individuals across data sources thereby making the identification of individuals possible with individual “quasi-identifying” information. Another less obviously personal challenge relates to who can access private data, and how to control access in the most secure way.

All of these challenges will need to be met before the potential for Big Data in transportation can be harnessed. As such, this paper aims to provide an in-depth awareness of the large-scale implementation challenges currently facing the use of Big Transportation Data in design, planning, and operations of transportation. It also provides insights into how we believe these challenges are likely to be met.

The paper continues with a section describing the scope of this paper and moves on to define Big Data, Big Transportation Data and from where they come. The next section describes the current state of the transportation literature as it relates to Big Data. This is followed by a background section on system architecture needed to understand the sections on the four main challenges to the widespread use of Big Transportation Data in transportation planning. A concluding section sketches our understanding of how the challenges of Big Transportation Data are likely to be overcome in the future.

**Scope of This Work**

The four large-scale challenges to the widespread use of Big Transportation Data identified in this paper have resulted from a thorough literature review. Since there is very little attention to this question in the transportation literature, most of the literature reviewed has come from computer science, computer engineering, and fields the most advanced in the use of Big Data, such as health and agriculture. The primary Google Scholar search terms used were “Big Data implementation challenges” and “Big Data technologies”. Relevant papers from articles resulting from these searches were then included in the literature, and this process was done iteratively. The more than 150 papers resulting from this process were placed into four categories of challenges: storage, processing, integration, and data privacy. These challenges concentrate on those relating directly and uniquely to Big Data. While other challenges such as data security, integrity and transfer are relevant to Big Data, they are not unique to Big Data, and so we do not concentrate on them here. Interested readers can consult the vast literature on these topics elsewhere (Tankard 2012; Tierney et al. 2012; Lagoze 2014). We continue by defining both Big Data and Big Transportation Data, as well as from where they come.

**Key Characteristics of Big (Transportation) Data**

Big Data has been described, characterized and defined in both academic and non-academic (traditional media, trade press, etc.) sources. Across these sources, there is a great variety in how Big Data has been defined and characterized (McAfee et al. 2012; Hashem et al. 2015; Zikopoulos et al. 2011; Wu et al. 2014). Often, Big Data are characterized by words beginning with the letter “v”. One problem with such “v-words” is that there is often variation in how they are defined from one author to another. Also, “v-words” do not necessarily define characteristics of only Big Data but of “non-Big-Data” as well. Finally, there are some concepts critical to understanding the challenges for the widespread use of Big Data that are not easily described with “v-words”. Given the confusion around definitions and the fact that we are most interested in the characteristics of Big Data as they relate to the challenges of using it, we discuss two types of characteristics, not all of which are “v-words”. As such, below we discuss “defining” and “non-defining” characteristics of Big Data.

**Defining Characteristics of Big Data**

Defining characteristics of Big Data are those that are unique to Big Data as opposed to data in general. Those critical to understanding the challenges of widespread use of Big Data are those from the most-cited definition of Big Data by the Information Technology (IT) advisory firm Gartner. According to Gartner:

“Big Data is high-**volume**, high-**velocity** and/or high-**variety** information...”. (Gartner 2012).

**Volume** refers to the size of individual datasets. Already in 2011, there were 2.5 quintillion bytes of data created...
every day (Hilbert and Lopez 2011), and this number keeps increasing exponentially (Jagadish et al. 2014; Kahn 2011), so that “Big” datasets currently typically range from zettabytes ($10^{21}$ bytes) to yottabytes ($10^{24}$ bytes) (Chen and Zhang 2014). It is often said that “Big” datasets are too large to be handled by an individual computer (Katal et al. 2013).

Velocity refers to the rate at which data are being generated. As with volume, the figures on rates of data being produced and received can be staggering. It was reported in March 2018 that over 900 million photos were uploaded to Facebook (Gewirtz 2016). In addition to being related to the rate with which data are generated, velocity also encompasses a notion sometimes referred to in the literature as variability (Gandomi and Haider 2015). Whereas velocity refers to the rate at which data are generated, variability refers to variance over time in data flow rates.

Variety refers to the structural heterogeneity of data. That is, data provided in different formats, some structured and others not. Structured data are mostly in the form of tabular schema-imitating spreadsheet and relational database systems. Text, audio, images and videos are examples of unstructured data, with Extensible Markup Language (XML), being an example of semi-structured format (Gandomi and Haider 2015). Unstructured data are more difficult to process, store and integrate and is becoming more common (Mansuri and Sarawagi 2006; Choi et al. 2006; Doan et al. 2009).

**Non-defining Characteristic of Big Data**

A non-defining characteristic of Big Data is simply one that applies to other types of data as well. Such characteristic creates an important challenge for the widespread use of Big Data, as Big (and non-big) Data are either by nature personal, or can be personal. By personal we mean that an individual’s identity is explicit, or can be revealed. That data can be personal we mean that different data sources can be combined to identify an individual and other information about them. While this is not a new problem [e.g., it has been a concern for a long time with traditional census data (Samarati 2001)], it becomes compounded with Big Data. This is so because of the many potential different sources of data available on people (Doan et al. 2009) and also because of the very personal nature of some Big Data (e.g., precise location data, medical records, etc.) (Xu et al. 2014).

Recently, large data collection organizations (i.e., government, institutions and non-governmental organizations) have begun adopting “open data” initiatives that allow for data to be freely available, shared, redistributed and reused by the public without restrictions of use (Auer et al. 2007). As such, open data can serve as a resource for private, public and academic research. The availability of such data means privacy has become of even greater concern.

**Big Transportation Data**

We characterize Big Transportation Data (BTD) simply as Big Data (as characterized above), but with potential transportation system applications. That is, data that could be used in areas in the traditional purview of transportation design, planning and operations, such as travel demand forecasting, infrastructure planning, transit network planning, operation optimization, etc.

**Where Does Big Transportation Data Come from?**

BTD comes from the combination of three types of technologies. We begin with two broad categories of devices that collect BTD; location-ignorant and location-aware devices.

Location-ignorant devices are able to sense the presence of other devices, although they are not explicitly aware of their own locations. These include technologies such as Bluetooth (Perego et al. 2017), Wireless Fidelity (Wi-Fi) (Perego et al. 2017), Global System for Mobile (GSM) (Perego et al. 2017) and closed-circuit television (CCTV) (Gill and Spriggs (2005)).

The second are devices that can determine their own whereabouts, i.e., they are location-aware. These devices typically derive their locations based on the location of other devices such as Wi-Fi routers, GSM towers, or satellites part of various navigation satellite systems, such as the Global Positioning System (GPS). They include GPS units, GPS navigators and most importantly smartphones.

While devices that collect data are critical for being able to use BTD, its potential can only be harnessed if the devices are connected to a communications network, such as the Internet (Stamp 2011), private local area networks (LAN) (Stamp 2011) or wide-area networks (WAN) (Stamp 2011). These networks allow the transfer of data from collecting devices to database storage systems from where they will be accessed for processing and analysis by end-users. Figure 1 provides a schema of the BTD ecosystem.

**The Current State of BTD in Transportation**

The combination of location-ignorant, location-aware and communications networks has led to the birth of Big Transportation Data. Academia as well as the public and private sectors have not overlooked the potential for BTD in transportation.
Research with Data Collected with Location-Ignorant Devices

In recent years, academic research has been conducted with the use of data from location-ignorant devices in public transit planning and operations. Transit smartcard data has been at the forefront of this to understand travel behavior (Bagchi and White 2005; Pelletier et al. 2011) and transit user loyalty (Trépanier and Morency 2010), also state that smartcards can be used to ascertain the loyalty of transit users in a network. Wi-Fi network data have also been used to understand (primarily pedestrian) travel behavior based on connection histories to wireless routers (Poucin et al. 2016; Shlayan et al. 2016). Similarly, Bluetooth receivers have been used to assess automobile route choice and travel times on alternate routes (Hainen et al. 2011).

Research with Data Collected with Location-Aware Devices

Location-aware technologies have been developed to determine their own location. Location sensors derive precise locations through the use of GSM, Wi-Fi and GPS (Leick et al. 2015; Van Diggelen 2009). Transport operations, planning and research heavily rely on these devices for precise spatiotemporal data in analysis and decision-making. Location-aware technologies is discussed in two categories namely GPS and smartphones.

GPS

Navigation GPS devices have long been used for finding the location of point of interest (POI). Transportation fleet operations rely heavily on navigation GPS systems that provide mobility trajectories of fleets. Much academic research has been done to cover the application of navigation GPS devices in transportation. Davies et al. (2010) evaluated the use of GPS devices for providing location-aware visual and auditory prompts for people with intellectual disabilities to enable them in navigating bus routes. Handheld GPS devices have been extensively used for travel mobility surveys in research (Draijer et al. 2000; Stopher and Greaves 2007; Montini et al. 2015). A study on children’s mobility using GPS-tracking device and mobile phone survey was conducted in Copenhagen (Mikkelsen and Christensen 2009). The research shows diversity of mobility patterns for children and the geographic inter-dependency of child mobility. Surveying and data collection with navigation GPS devices are becoming phased out due to the emergence of location-aware smartphones that assure precise location from satellites and can augment location from cell phone towers in places with poor satellite signals.

Smartphones

Pervasive smartphone devices have gained popularity recently for mobile and internet communication. Many mobile applications (e.g., social media, maps, dating apps, locations and others) are used daily on smartphones by their
users. Location-aware applications are common in smartphones, they observe the location of the user and report to a location-based service (LBS). Location-based services provide queries of point of interest within a defined proximity of the user as reported by the smartphone. As an example, a smartphone user can ask (query) for restaurants nearby or within a distance of his/her current location to receive a list of matched restaurants. Smartphones have inbuilt assisted-GPS sensor for precise location tracking to satellites, in cases where cloud visibility is achieved. At places with less cloud visibility, smartphones can gain location by connecting to nearest cellphone towers or Wi-Fi access points. A large body of literature has contributed to the use of smartphone in transportation studies.

Patterson and Fitzsimmons (2016) conducted an experiment on participants from Concordia University that used a smartphone travel survey developed to collect passive data on human mobility whilst minimizing the respondent burden. Respondent burden is reduced in such surveys relative to traditional self-reported surveys. An enormous amount of location-sensitive data is gathered on social media platforms like Facebook, Twitter, Instagram and others.

Information Inference from BTD

Another research area receiving attention in the transportation literature is that related to the development of methods allowing the inference of the main aspects of transportation demand required for traditional trip-based transportation demand forecasting. As such, data inference methods have been developed in the following areas. The inference of trip ends was one of the earliest questions to be broached in the literature (e.g., Stopher and Greaves 2007), but one which continues to have (primarily rule-based methods) methods developed [e.g., (Patterson and Fitzsimmons 2016; Zhao et al. 2015)]. Mode detection has received the greatest amount of attention in the literature with methods evolving from rule-based (e.g., Bohte and Maat 2009) to discrete choice (e.g., Bierlaire et al. 2013) and machine-learning approaches (Gonzalez et al. 2010; Reddy et al. 2010).

Purpose detection has turned out to be the most difficult to infer. Initial rule-based (e.g., Wolf et al. 2001) continue to be used (e.g. Shen and Stopher 2013), but are being replaced with machine learning algorithms (e.g., McGowen and McNally 2007; Griffin and Huang 2005) increasingly using data collected from various BTD sources such as social media (e.g., Yazdizadeh et al. 2018).

Finally, itinerary inference has evolved from simple map matching methods (see White et al. 2000) to more sophisticated probabilistic approaches (Bierlaire et al. 2013). Itinerary inference has been applied primarily to road networks and particularly to automobiles (e.g., Bierlaire et al. 2013) and bicycles (e.g., Hood et al. 2011). Less common are methods for inferring transit itineraries combining smartphone and GTFS data (Zahabi et al. 2017).

Future Sources of BTD

In addition to current sources of BTD, we also have to include the coming addition of autonomous vehicles as a data source. According to Intel (Krzanich 2016), the evolution of autonomous vehicles (AV) with their on-vehicle sensors and cameras will generate and require enormous amounts of data. AV cameras alone will generate 20–40 Mb/s per vehicle, while radars will generate between 10 and 100 Kbps with an estimated average of 40 terabytes of data for every 8 hours of driving (Krzanich 2016).

Summary of Current BTD Research

As can be seen from the rest of this section, there is a great deal of research being done on BTD in transportation. Collectively this work can be divided into three broad categories. The first category relates to the use of various technologies in actual data collection (Arentze et al. 2000; Efthymiou and Antoniou 2012). The second category concentrations on challenges related to methods that process BTD and seek to infer information from it that can be useful in transportation (Yazdizadeh et al. 2018). The third category focuses on the evolving technologies that present opportunities for the successful implementation of BTD. While this work is clearly necessary for BTD to be effectively used in transportation, there has been little emphasis on the importance of system architectural components necessary for large-scale adoption of BTD.

System Architectural Components

Critical to understanding the challenges of BTD is an understanding of data system architecture more generally. Data management architectures (DMAs) organize the flow of data from collecting devices to the storage systems with which data are managed. DMAs can be split into three essential elements. First is the physical infrastructure (i.e., hardware) needed to be able to store data. Second are file systems with which files (and their underlying data) are organized on hard drives. Third are database management systems.

Hardware

We begin with the hardware side of data management systems and with data retrieval. Data retrieval typically, and traditionally, involves an in-between step; data must be read from long-term storage on hard drives into active memory. The speed with which this happens is dependent upon three
elements: computer processor (CPU), disk characteristics, and disk connection to active memory. The faster the processor, the faster data can be read into active memory (Abadi 2016; Ousterhout and Dougis 1989). Disks themselves vary in the speed with which data can be accessed from them. Traditional spinning hard disk drives (HDDs) have slower transfer speeds than solid-state drives (SSDs), from which data can be accessed directly from its storage sector (Tsirigianis et al. 2009). Finally, the connection between hard drives and active memory plays a critical role in the speed with which data can be accessed; see Fig. 2.

Transfer speeds are fastest from directly attached storage (DAS) (i.e., hard drive on a single node, such as a server or other standalone computer). Speeds decrease with a greater separation of where the data are stored and active memory with network attached storage (NAS) (i.e., connected through a local area network (LAN) having slower speeds than DAS, and storage area networks (SANs) (e.g., storage on remote networks) potentially taking even longer than LANs (Abadi 2016; Patil 2016). The writing of data to storage involves the reverse process, i.e., from active memory to final storage.

**File Systems**

On hard drives, data are stored hierarchically. At the lowest level, data are stored in a binary format as bytes with a location on a hard drive (Abadi 2016; Ousterhout and Dougis 1989). Bytes are grouped together as “data” (e.g., the content of a spreadsheet cell) and data are grouped together into files. There are different underlying logical systems by which bytes can be organized into data, and data into files. These logical systems are known as “file systems”, that are a subsystem within the operating system (e.g., Linux, Windows, MacOS, etc.) (Ousterhout and Dougis 1989; Tanenbaum and Woodhull 1987). There are many file systems that exist, but the most common are NTFS, VFAT, EXT3 and HPFS (Tanenbaum and Woodhull 1987).

**Database Management Systems**

While file systems hierarchically organize data and files on hard drives, database management software uses the file system to make data available for processing. This is done with database software. The traditional and most popular database software products are based on structured query language (SQL). SQL resulted from the work of E.F. Codd who introduced the “Relational Model” in the 1970s (Codd 1970). As a result, these products are also known as relational database management systems (RDBMS) of which there are many examples (e.g., MySQL, PostgreSQL, Microsoft SQL Server, Oracle DB). RDBMSs, now typically referred to as “legacy” systems, have proven very efficient for intensive amounts of data storage, retrieval and processing for many decades (Vicknair et al. 2010). RDBMSs are organized into databases containing tables, with tables related to each other by common identifier constraints (i.e., keys). Database table schemas are strictly defined. That is, data can only be read into them if it adheres to the structure defined in the schema (e.g., text data cannot be read into a variable defined as an integer). The structure placed upon the data is a primary factor making such systems so efficient at saving and accessing data. Also, RDMBSs are typically “centralized” meaning they are deployed on one node and cannot be easily scaled to multiple nodes.

Finally, RDBMSs are “transactional” (Gray and Reuter 1992; Maier 1983) which means they also demonstrate the following properties. First, Atomicity guarantees that all transaction operations are executed “all-or-nothing”; if one part of a query fails, the entire query fails and none of it

---

Fig. 2 Disk storage drives
is executed. Second, transactional consistency guarantees every transaction will bring the database from one valid state to another. Third, Isolation ensures concurrent transactions (e.g., from multiple users) will be executed sequentially. Fourth, Durability ensures that once a transaction has been committed, databases remain the same in the event of a power loss, system error, crash, etc. Collectively these four characteristics are known as “ACID” properties of a transaction (Maier 1983).

Challenges and Opportunities in “Storing-It-All”

The first challenge identified in the literature is to actually being able to store and manage all the BTD. This concerns the “v-word” “volume”. The volume of data that will need to be stored is a challenge for using Big Data in general, but is clearly also a challenge in transportation in particular with the many new sources of data (described in “Where Does Big Transportation Data Come from?”) available with transportation applications. As an example, it is now possible to record mobility traces collected by cell phone operators, traffic information, transaction systems (integrated ticketing, road user charging, car park payment, electronic fee collection), cameras, in-vehicle GPS, social media and smartphone geolocation technologies (Chen and Zhang 2014). The rich data gathered from these sources will help to improve on transport modeling and planning to deliver accessibility, efficiency and economic performance potential which hitherto was not possible. Ultimately, this boils down to adding capacity; faster CPU, hard drives with more storage capacity from which data can be accessed (and written) more quickly, and enabling software. Such capacity can be added in two ways; vertically, or horizontally (see Fig. 3).

Vertically Scaled Systems

The traditional approach to increasing data storage and management capacity is “vertical scaling”. This involves improving the capacities of a single node (i.e., a standalone computer). Since traditional RDBMSs were designed for deployment on such systems, there are few software implications and as a result, vertically scaling concerns primarily hardware. As such, it entails the use of faster CPUs, the increase of active memory (RAM) and the addition of larger and faster disk drives (e.g., converting from HDD to SSD) as shown in Fig. 3.

While hardware improvements lead to vertical scaling, there are limitations to just how “high” such systems can be scaled. While Moore’s law suggests increasing improvements in CPU speeds, we are limited to the available chip technology at any given time (Ousterhout and Douglish 1989; Schaller 1997), even when considering the possibility of multiple cores on the same node. Secondly, there is no guarantee that Moore’s law will continue into the future (Ousterhout and Douglish 1989; Kish 2002). Similarly, capacities are limited by available active and long-term storage technologies. Moreover, it may be possible to scale up to required capacity with available technology in some circumstances, but component cost increases dramatically with improvements at the cutting-edge of performance. Finally, vertically scaling, a single node, amounts to putting all of your eggs in one basket, the downside of which is that if there is a problem with the vertically scaled node (e.g., it crashes), data cannot be read or written. In other words, vertical integration increases the risk of greater downtime.

Horizontally Scaled Systems

Instead of increasing the capacity of a given node, horizontal “scaling-out” involves the combination of different nodes

Fig. 3 Scaled systems (systems sizes are for illustration purposes only)
into a “cluster”. That is, a “distributed” storage system. As illustrated in Fig. 3, nodes with similar (homogeneous) or varying (heterogeneous) capacities are added to the cluster to meet storage and computing needs.

Distributed systems have the following advantages compared to single-node systems. First, it is possible to add resources (CPU, active and long-term memory) in a cost-effective manner since capacity can be increased almost limitless without the skyrocketing costs associated with performance increases in a single-node.

Second, distributed systems typically store redundant copies of data across multiple nodes, which decreases the risk of data not being available at any given time. The storage of multiple copies is done in the following ways. The same data can be stored on different nodes. This, referred to as “redundancy”, means that if one node goes offline, the data are still available on another node. Additionally, data can be “sharded”. This means that different parts of the same dataset can be stored on different nodes. For example, the columns (or rows) from the same database table can be stored separately, thus increasing the speed at which data can be accessed and written.

As with vertically scaled systems, database software is required for the proper functioning of horizontally scaled systems. At the same time, the limitations of traditional RDBMSs make them inappropriate for horizontally scaled systems. A key characteristic of horizontally scaled systems is that data are synchronized across nodes within the system. Traditional RDBMSs were not initially designed with this in mind, so they remain relatively inflexible in this respect making synchronization with them inefficient and arduous (Moniruzzaman and Hossain 2013; Vaquero et al. 2011). This inflexibility is ultimately due to the reliance of RDBMSs on traditional, centralized file systems (see “File Systems”). Such file systems do not easily allow the management of files across multiple nodes.

As a result, horizontal scaling requires both hardware in the form of nodes and networks, as well as distributed database management systems (DDMS) that are designed to seamlessly synchronize data across nodes. In order to do this, DDMSs themselves rely on non-centralized distributed file systems. DDMSs and files systems make up the software component of horizontally scaled systems (Vaquero et al. 2011).

**Distributed File Systems**

The logical hierarchy of centralized file systems locates bytes on a single hard drive and groups the bytes into data and files. Distributed file systems on the other hand use a slightly deeper hierarchy. Bytes are stored on a hard drive, organized into data, data are organized into “chunks” and chunks into files (Ousterhout and Douglass 1989). Chunks themselves, however, do not have to be stored on the same hard drive. So, in addition to a deeper logical hierarchy, the key feature of distributed file systems is that they can also locate data across different hard drives. While several distributed file systems exist, the most common are the Google file system (GFS) and the Hadoop File System (HDFS).

GFS, developed by Google Inc. (Google 2018) supports large-scale and data-intensive applications (Ghemawat et al. 2003). It can be deployed on any standard node thus making it desirable from a cost perspective when scaling-out a system. The distribution of chunks across hard drives with GFS is orchestrated by one “master” node to the sub-nodes (“slaves”) of the system. This organization means that if the master node goes offline, access to data on the master and slave drives becomes impossible. As such, GFS is said to have a single point of failure.

The Hadoop File System (HDFS) (Shvachko et al. 2010), designed by Apache like GFS also runs on any standard node and is suitable for data-intensive applications. It is also based on a “master–slave” architecture, and as a result also has a single point of failure. Compared to GFS, HDFS has become much more common in industry application, and has had a series of DDMSs built using the underlying HDFS (Shvachko et al. 2010; Borthakur 2007; Shafer et al. 2010).

**Distributed Database Management Systems**

In addition to specialized file systems, and due to the limitations RDBMS, horizontally scaling also requires dedicated database management systems (DDMSs). A number of such systems exist and fall into broad categories; structured and unstructured. Basically, such systems are distributed versions of RDBMSs. That is, they allow for the distribution and synchronization of data across multiple nodes, but they remain structured database systems. The most common such systems in use are Google Big Table (Chang et al. 2008) and Apache HBASE (Vora 2011). Another increasingly common DDMS is NoSQL, which in addition to being designed for horizontally scaling is also unstructured.

**Characteristics of Horizontally Scaled Systems**

In order to be effective, horizontally scaled systems need to be planned well. Key characteristics of effective distributed systems have been summarized in Brewer’s CAP Theory (Brewer 2000; Gilbert and Lynch 2002) (see Fig. 4):

Consistency (C) While redundancy means having multiple copies of the same data in different locations, “Consistency” means that all copies of redundant data are identical (Oracle 2015).

This ensures that the most up-to-date data are available even if there are server or network failures.
Availability (A) Distributed systems operate on multiple nodes that run concurrently in the implementation of a task. As a result, individual nodes can stop operating (e.g., due to a crash failure). Such failures are common and inevitable in networked systems. Availability means there is a sufficient number of nodes with redundant data that all data can be accessed at all times, even if one or multiple nodes crash (Oracle 2015).

Partition tolerance (P) Partition tolerance is similar to availability in that it describes systems where redundant data can be accessed at all times. With partition tolerance, however, the concern is not with nodes themselves, but with the network connectivity of the nodes (Fathi 2013). This can be seen as “network availability”.

While ideally, distributed systems would have all three of these characteristics, in practice they are typically characterized by two at most, with system design amounting to trading-off between the characteristics (Gilbert and Lynch 2012). While systems that are not distributed over different networks exist, discussion on distributed systems is typically limited to those that are. As a result, we describe only systems demonstrating partition tolerance that is AP and CP systems.

AP systems are characterized by availability and partition tolerance. Such systems are made up of multiple networks (P) with a node (or cluster of nodes) (A) on each network. Additionally, each node (or cluster) would be able to operate without communication to the others. If communications between the nodes/clusters were interrupted, updates to data would be out of sync and as such, the system is not always consistent (i.e., it does not demonstrate strong consistency). Once all networks are functioning, data will become synchronized again but with delays (eventual consistency). Well-known “AP” systems include CouchDB and Cassandra (see “NoSQL and NewSQL” “NoSQL and NewSQL” below).

CP systems are characterized by consistency (C) and partition tolerance (P). Such systems are made up of multiple networks (P), but with only single nodes on each network. CP systems maintain multiple copies of the same data and therefore are “Strongly Consistent”. Unlike AP systems, if there is a network failure, there is always sufficient network redundancy, that the data across all nodes remains consistent. At the same time, since there is only one node per network if one of the nodes fails, there is no node redundancy, and as a result, the system is not “Available”. Well-known “CP” systems include MongoDB and Redis (see “NoSQL and NewSQL” “NoSQL and NewSQL” below).

As such, the volume of BTD presents a major challenge to the potential to use it effectively in transportation in the future. At the same time, new approaches and technologies, namely the use of scalable distributed systems appears to be the most probable solutions to meeting this challenge, with the design of the systems requiring choices and trade-offs to be made between consistency, availability and partition tolerance.

Data Storage Opportunities for Transport Systems

The recent advent of sensor-based technologies such as infrared detectors, video detectors, induction coils at bayonet points, laser detectors and others, for real-time traffic monitoring and the passive data collection of mobile user trip data for transport modeling (i.e., mode and activity inference) contribute a rich dataset for real-time analytics and decision-making by transport stakeholders. In this regard,
Challenges and Opportunities in Unstructured Data Storage

The second challenge identified in the literature is being able manage BTD of many different data formats. This concerns the v-word, “variety”. As with data volume, this is a challenge to using Big Data more generally, as well as BTD.

In general, data can be formatted on a continuum between structured and completely unstructured data. Structured data (described in “Database Management Systems”) is highly organized and format schema are defined before data are even collected (i.e., before it is stored in a database). In fact, if structured data are expected for a relational database but the data are not sent in the pre-determined format, it will typically not be stored at all. On the other end of the spectrum is unstructured data. Unstructured data are negatively defined as that not adhering to any predefined data schema. It comes in two main types; text and non-text. Examples of unstructured text data are email messages, text documents, etc. Examples of non-text unstructured data are satellite images, CCTV videos, etc. In between structured and unstructured data there also exists semi-structured data. Semi-structured data encapsulates unstructured data within a meta-structure using semantic tags and marking. Common semi-structured formats include mark-up languages (e.g., HTML, XML) and JSON (Java Script Object Notations). Different formats present two major challenges. First, mechanisms are required to be able to save and access the data in an efficient manner. Recall that structure in traditional RDBMSs is what allows them to efficiently manage large amounts of data. Second, taking advantage of BTD also means taking advantage of different sources of data, typically in different formats, so integration of the different data sources is a challenge. Being able to use data of different formats ultimately requires the use of software that can accommodate a variety of formats in a structured manner that also allows efficient retrieval. The most common DDMSs rely on frameworks based on NoSQL (Moniruzzaman and Hossain 2013; Pokorny 2013) with NewSQL being a more recent and quickly evolving framework.

NoSQL and NewSQL

NoSQL databases (i.e., non-formally structured relational databases) are becoming more popular for Big Data storage. NoSQL databases are much more flexible allowing the following features that are impossible in RDBMSs: the ability to add new variables and modify existing variables within tables, without the need to drop and recreate tables; support for copying and pasting data into and from tables; more flexible integration of different programming platforms through Application Programming Interfaces (APIs); eventual consistency (see “Characteristics of Horizontally Scaled Systems”), and supports the management of data across nodes and in quantities too large for one node. At the same time NoSQL systems are not transactional, and as a result do not demonstrate ACID properties (see “Database Management Systems”). NoSQL databases are becoming the core technology for Big Data and can be characterized according to one of four data models: key-value, column-oriented, document-oriented, and graph. We describe these models below.

In key-value databases each observation (row) is stored as a dictionary, with each key defining a variable. Queries can be made directly according to keys. Such databases are characterized by high expandability (easy to add or remove variables without having to create new tables) and shorter query response time than those of relational databases. These databases have suitable storage structure for continuously growing, inconsistent values of Big Data for which faster response of queries is required. Key-value databases provide support to large-volume data storage and concurrent query operations. Popular examples of key-value NoSQL DDMSs are MongoDB (Dirolf and Chodorow 2010), Cassandra (Lakshman and Malik 2010) and DynamoDB (DeCandia et al. 2007).

Column-oriented databases store columns of data separately, unlike RDBMSs where data are stored in the form of complete records. They are suitable for vertically partitioned, contiguously stored, and compressed storage systems. Reading of data and retrieval of attributes in such systems is quite fast and less resource intensive than RDBMSs, as only the relevant column is accessed and concurrent process execution is performed for each column (Abadi 2016).
Column-oriented databases are highly scalable and eventually consistent. Examples of column-oriented DDMSs are HBase (George 2011) and HyperTable (Khetrapal and Ganesh 2006).

Document-oriented database are similar to key-value DBs and store data in the form of key and value as reference to a document (i.e., a file). However, document databases support more complex queries and hierarchical relationships. This data model typically uses the JSON format and offers very flexible schema (Chodorow 2013). Although the storage architecture is schema-less for structured data, indexes are well defined in document-oriented databases. SimpleDB is the only database that does not offer explicitly defined indexes (Cattell 2011; Calil and dos Santos Mello 2012). Document-oriented databases extract metadata to be used for further optimization and store it as documents. CouchDB (Anderson et al. 2010) and SimpleDB (Chaganti and Helms 2010) are two examples of document-oriented DBs.

Graph databases are extensions of key-value databases. As such, each observation (row) is stored as a dictionary or a series of nested dictionaries (primarily in JSON format). The nested dictionaries contain relational structure. Graph databases offer persistent storage of objects and relationships and support simple and understandable queries with their own syntax (Iordanov 2010). This allows data to be linked together directly, which can be accomplished with one operation making querying more efficient. Modern enterprises are expected to implement graph databases for their complex business processes and interconnected data, as this relational data structure offers easy data traversal (N. Developers 2019). The most common Graph DB is Neo4J (Vukotic 2015).

Finally, NewSQL is an emerging DDMS technology that extends NoSQL approaches while building upon attractive features of traditional RDBMSs. Whereas NoSQL does not provide ACID guarantees for database transactions, NewSQL approaches do. As a result, NewSQL approaches combine the best of traditional RDBMSs and NoSQL approaches. At the same time, NewSQL are rapidly evolving and do not always have extensive support. As a result, we mention them as an avenue of considerable potential, but which remain in development and an interest for research (Chen et al. 2016; Stonebraker 2012; Grolinger et al. 2013). The most popular NewSQL frameworks are NuoDB (Brynko 2012), VoltDB (Stonebraker and Weisberg 2013), Google Spanner (Corbett et al. 2013) and CockroachDB (Corbett et al. 2013).

Opportunities for Unstructured Transport Data

Evolving transport systems ingest data in the formats of images, videos, audio and various other unstructured data formats. As a result, ITS architectures need schema-free databases to store non-related data provided by traffic surveillance and traffic sensor systems, which hitherto could not be stored in traditional RDBMSs. Orru et al. (2017), however, built an ITS application with a backend of a NoSQL database to create a public access of public transport information (of GTFS files) all over the world and also search for geotagged photos. NoSQL systems allow for the storage of such schema-less files, which would be difficult to implement in a traditional database. Typically, travel mobility datasets are designed with varying questions (i.e., fields) based on the purpose of the survey that can contain unstructured formats like audio and images. NoSQL databases allow for the efficient storage of travel mobility data. Vela et al. (2018) focused on the design and storage of accessible transport routes, obtained by means of crowd-sourcing techniques, in a NoSQL graph-oriented database. The authors adopt a graph NoSQL database to address the integration of accessibility data from three sources, namely: existing open data, private data concerning actual accessible routes obtained through crowd-sourcing, and data from existing traffic sensors. NoSQL databases embrace the capability of a seamless integration of varying and non-related data, which is common in transport systems.

Challenges and Opportunities in Processing

The third challenge identified in the literature is being able process all of the BTD. This concerns the v-word, “velocity”. While processing is required in the management of data (i.e., storage), the main processing challenge is making use of collected data. The methods used to process data are a function of how quickly the processing is required, i.e., whether information is required in real-time or not. There are in general, two approaches to processing BTD: Batch (ex post) Processing, and Stream (real-time) Processing. These approaches require implementation using different Processing Engines, or Frameworks. Below we describe the approaches as well as the most common implementing Frameworks.

Batch Processing

Batch processing is the processing of large, complete, static or historical data sets, and provides information after the entire dataset has been collected (Chen and Zhang 2014; Moniruzzaman and Hossain 2013; Ji et al. 2012). In other words, results are not provided in real-time. As an example is OD surveys are conducted until completion of data collection before processing of data aggregation is done.

This approach is mostly adopted when processing finite (or bounded) datasets that are complete, whose size can be estimated, and that are persistently stored on a hard drive.
That is, the dataset is unchanging when it is analyzed and includes information for a given period of time (e.g., data from a regional OD survey). The data need to be complete because the types of calculations done on them require having all of the relevant data, such as when calculating totals and averages. In such situations, datasets must be treated holistically instead of as a collection of individual records. Also, the operations require that the dataset be unchanged for the duration of the calculations. Most common framework for batch processing is Apache Hive (Thusoo et al. 2009).

**Stream Processing**

Whereas Batch Processing requires datasets to be complete and static, Stream Processing systems operate on data immediately as it arrives (Chen and Zhang 2014; Ranjan 2014). As such, the data being processed do not need to be complete or static. Moreover, the size of the “entire” dataset is unknown at any given time until data are no longer collected, i.e., it is “infinite”, and its size is irrelevant for Stream Processing. To understand Stream Processing, it is useful to understand Stream Processing workflow.

Typically, in a Stream Processing environment, data are received continuously (although not necessarily at a continuous rate), and the data contain information that are not required for the immediate analysis for which results are sought. As such, a first stage of processing is to retain only data relevant to the processing goal. None of the other data is kept or stored. Once data are filtered, processing operations are done on individual observations “one at a time”. Stream Processing is well-suited to situations in which results are required in real-time.

An excellent example of situations requiring Stream Processing is Uber, the peer-to-peer ridesharing company (Uber 2018). Uber needs to analyze the location of its riders and to match them with the nearest drivers. They also need to determine the most efficient itinerary for the driver to the rider’s origin and destination once picked-up. Moreover, information on the location of the driver needs to be provided to the waiting rider. Once a trip is completed, Uber needs to calculate the cost of the trip and send this information to the rider. All of this requires processing to be done in real-time. An emerging technology for which Stream Processing is already required, and for which it will be required in greater amounts in the future is that of Autonomous Vehicles. While Uber needs to be able to process streamed data quickly, Autonomous Vehicles need to process information (read in data, react) instantaneously.

As with Batch Processing, specialized Processing Frameworks are required for Stream Processing. Also, as with Batch Processing, many such frameworks exist, with the most common being Apache Storm (Marz 2013), Kafka (Thein 2014) and S4 (Neumeyer et al. 2010).

**Challenges and Opportunities in Cyber-Security**

The fourth challenge relates to the fact that BTD infrastructure needs to be secured from unauthorized access by an attacker. This challenge is related to ensuring transportation system components are securely protected to avoid vulnerabilities exposed for an adversary to exploit and also protect data as it is transmitted on communication channels. We continue to discuss the context of cyber-security in transportation and known vulnerabilities to be considered.

**Cyber-Security of BTD**

Recent dominance of high-resolution information gathering devices (i.e., Cameras, transponders, wireless routers) and social systems are on a path of fully connectivity known as “Internet of Things”. A large body of research and standards has evolved on mining rich data ingested by these interconnected devices. Intelligent transport systems, gain access to a wealth of information from interconnected data from GPS location tracking to traffic logs, that aid in public safety, disaster recovery and emergency response. As modern transport devices contain a network of networks made up of embedded communication methods and devices contain a network of networks made up of embedded communication methods and scope, issues of cyber-security are raised.

Whilst discussion on IT Security is a fundamental challenge to core IT implementation and not limited to Big Data implementation, a scope of Cyber-Security is worth considering as it can impact on the veracity (truthfulness) of the data harnessed on large-scale integrations. Cyber-Security protects against illegal or unauthorized access to information sources and their communication channels which can disrupt service availability for interconnected devices. There is a need for devices and generated data to be adequately secured against attacks, vulnerabilities and exploits. Potential vulnerabilities that could be exploited in transportation include unsecure vehicle-to-vehicle communication, unauthorized vehicle data interception, seizure of control systems like brakes or accelerators. As an example, a group of civic hackers deciphered and exposed the bus location system of Baltimore (Rector 2015). In 2016, San Francisco transit was hacked to give unpaid access to commuters for 2 days (Guardian 2016). It is evident that uncontrolled attacks and vulnerabilities can defame the
purpose of intelligent transport systems and incur unforeseen losses that can destroy system implementation. Key vulnerabilities that are of concern in Big Transportation Data implementation are discussed below.

Vulnerabilities of software applications Most common threat to security for Big Transportation Data is exploits undertaken in software libraries and bugs. Software packages and Operating system (or firmware) kernels usually expose vulnerabilities or system bugs that hackers can exploit to gain unauthorized access to control the system. Software updates, patches or fixes are periodically developed by software manufacturers to update known vulnerabilities mostly through automatic system updates. As transportation information systems encompass a wide suite of software components (i.e., web server, database, application framework), it is required system updates from trusted manufacturers are allowed and enforced to ensure a robust secured platform for information share.

Vulnerabilities of field devices BTD ingest high-volume data from dispersed sensor and pervasive devices which are mostly located in remote areas and far from routine supervision. These remote field devices such as traffic lights, cameras, road counter equipments are often in isolated public places and remain susceptible to tampering. Isolated field devices are vulnerable to tampering thus an adversary, who can alter the physical configuration of devices can compromise a system by gaining illicit access to its information source. It is important a level of surveillance is provided for field devices which are deployed in isolated environments.

Vulnerabilities of communication networks Communication devices create an enabling environment for data exchanges between interconnected devices. Network vulnerabilities are well known within wired and wireless network service. Such vulnerabilities allow an attacker to eavesdrop on data packets which are exchanged in the communication channel. Cellular networks, mostly wireless services, are known to be vulnerable to signal intercepts and other threats. Wi-Fi network vulnerabilities are very common in hacker communities, who gain access and exploit the network including devices that are connected. A network map is a sensitive information to an adversary who might be interested in exploiting a transportation system thus its detail should be treated with high confidentiality. Data encryption and cryptographic algorithms such as Data Encryption Standard (DES) algorithm, Rivest–Shamir–Adleman (RSA) are applied to data packets to perturb the data content as they are transmitted over network channels. The underlying transport layer is made secured by adopting secured communication protocols such as Transport Layer Security (TLS) and Secure Sockets Layer (SSL) which provides privacy and data integrity between communication nodes.

Challenges and Opportunities in Privacy Protection

Until now we have focused on challenges related to defining characteristics of Big Transportation Data, namely the three Vs. The fifth challenge relates to the fact that BTD often contains personal data explicitly, or personal information that could be revealed by combining or analyzing data that is not strictly speaking personal, i.e., “Personally Identifiable Information” (PII) (Tene and Polonetsky 2011; Schwartz and Solove 2011). In other words, the challenge is related to ensuring the protection of individual privacy with the use of BTD. This is not a challenge uniquely for BTD, and the challenge of privacy protection in the face of PII has been an issue for a long time (see e.g., Sweeney (2002) who experiments identifying personal information by linking voter registration data sets to medical records). As a result, we do not concentrate on the general question of privacy protection with PII as it has received a great deal of former attention [see e.g., (Schwartz and Solove 2011; McCallister et al. 2010)]. What is unique about BTD is the large amount of temporally and geographically precise location data that can be collected on people. As such, this discussion focuses on the protection of privacy in the context of what we refer to as “Personally Identifiable Location Information” (PILI).

An example is given by Anthony Tockar (Neustar Research 2018), a summer intern at Neustar, an information-analytics who showed how to extract the exact location and time that celebrities used cabs in New York City extracted from open New York City Taxi and Limousine Commission (TLC) data. By joining the two data sets, Tockar found the cash tips paid by celebrities (Neustar Research 2018).

Transportation planning agencies have had access to both PII and PILI in the past through routine data sources collected for planning purposes such as origin–destination surveys. As a result, they have used techniques to protect privacy both internally, as well as when such data are shared with third parties such as consultants and academic partners.

With greater amounts of or more detailed information about people, these methods will need to be adapted. Such adaptation is becoming increasingly important with open data policies (see e.g., Ville de Montreal 2018), which are becoming more common and which by their nature impose much less control on who and the number of people who have access to potentially identifiable information. An understanding of the techniques used for privacy protection in the context of PII, and available for use with PILI, requires an understanding of underlying data anonymization operations. We begin with these and then continue with a description of anonymization techniques as they have been applied with PII and how they are applicable to PILI.
Data Privacy and the Need for Anonymization

Information collected for transportation planning and operations purposes can contain “microdata”, i.e., detailed information on individuals and households (addresses, age, sex, etc.) (Ghinita et al. 2007; Cormode and Srivastava 2009). Data attributes (or variables) that identify individuals are referred to as “Explicit Identifiers”. Attributes that do not explicitly identify individuals or households can, in combination with other attributes, potentially identify record owners uniquely (Sweeney 2002). Such attributes (e.g., zip code, sex, date of birth, etc.) are referred to as “Quasi-Identifiers”. While being able to identify individuals is an issue in itself, it becomes even more critical when “Sensitive Attributes” (e.g., disease, income, etc.) (Cormode and Srivastava 2009) are available.

Another issue affecting privacy protection and concerns is to whom data are available. To best understand the issues surrounding this, we define what we refer to as the Data Chain of Custody (DCC) that describes how data pass from the individual on whom it is collected to the end user of the data. The Data Chain of Custody is an adaptation of Xu et al.’s (2014) data “User Roles”.

The chain begins with the data owner (same term as Xu et al.) who is the person on whom data are being collected. The owner’s information is recorded by the “Data Recorder” typically a device, such as a smartphone. The data collector arranges the collection, stores and curates the data for the data analyst. It can be an individual researcher, a governmental institution (e.g., regional planning authority) or a private company. Data collectors can collect data for their own purposes, or on behalf of others. Data analysts process, analyze and integrate collected data for the end user. Multiple roles can be played by the same individual or institution, so that for example the data collector might also be the data analyst and end user. Sometimes the data owner (in the case of location-based services) can also be the end user. We quickly provide three examples of BTD and the DCC.

The first example relates to the smartphone travel survey platform, Itinerum (Patterson and Fitzsimmons 2017). This platform allows researchers to develop and administer their own customized smartphone travel surveys (see e.g., Patterson et al. 2018). While the platform also allows some data processing, in this example, we assume that the survey administrator only uses it to collect data and does analysis in-house. As such, this example involves a municipality that undertakes a smartphone travel survey that it will use for analysis of their local transportation system, as the City of Montreal did in 2016 (Patterson 2017). In such a circumstance, the data owner is the respondent with their smartphone being the data recorder. The data collector is the Itinerum project that is collecting the data on behalf of the municipality. The municipality performs analysis on the data and therefore is the data analyst. Because the municipality will use the analytical results from the collected data, it is also the end user.

The second example is someone requesting a list of nearby restaurants through Google Maps on their smartphone, also known as a location-based query. In this case, the data owner is the person searching for restaurants and their phone the data recorder. Google is the data collector since it developed the app and infrastructure and stores the owner’s location data. Google is also the data analyst since it processes the request and returns a list of nearby to the user. As such, the owner is also the “End User”; see Fig. 5.

Lopez and Farooq (2018) propose a transportation blockchain system to protect the personal travel information and improve the privacy of respondents to passively solicited data.

The proposed system protects users by making them the data owners and controllers of their personal information and is secured by a private key which can be accessed through smart contracts. The blockchain performs the role as a data collector by assigning keys, maintaining a transactional ledger and smart contracts to the information which the data owner seeks to share. Data analyst, mostly third parties, require a smart contract to access travel information.

Fig. 5 Dataflow across data agents

Springer
Data privacy risks are related to the DCC and in particular who, and under what circumstances, has access to the data. A data privacy breach results when someone’s identity (possibly associated with Sensitive Attributes) is revealed in a dataset when it is not supposed to be. This can happen unintentionally and with no malicious intent. When it happens intentionally and with malicious intent, it is referred to as “Adversarial” (Hasan et al. 2013; Lindell 2005).

As the number of people accessing data, and the number of people accessing data whose identities are not known, increases, so does the risk of adversarial data privacy breaches. When data are available to few known individuals (e.g., to data analysts in a municipal planning agency), privacy risks are limited. This is because the people with access are known and typically employees operating under regulations. Also, fewer people accessing data implies lower probabilities of discovery of private information that could be revealed when combining data sources, quasi-identifiers, etc. This situation is at one end of the privacy risk spectrum with open data being at the other.

With Open Data, there are unknown numbers of unknown people accessing data. So, the characteristics of data and the degree to which data are available to known or unknown users determine the risk of the revelation of private information. Privacy protection with PII is implemented with a number of different anonymization operations, which are applied in different combinations in Anonymization Techniques (or Anonymization Models). We first discuss Anonymization Operations and then Anonymization Techniques.

### Anonymization Operations

The most popular anonymization operations used in application are: generalization, suppression and perturbation. Generalization performs anonymization on data by replacing some values of an attribute with a taxonomy of its parent value (Nergiz et al. 2008; Bayardo and Agrawal 2005). A set of attributed values are replaced by a general categorical description value (e.g., replacing language spoken at home with English or Other). Generalization operations are mostly applied to quasi-identifiers and sensitive attributes, and reduce the probability of uniquely identifying a record owner. A numerical interval or range is typically used to generalize numerical attributes. Specialization is achieved when generalization is reversed by returning the detail of specific values.

Whereas Generalization works with taxonomies, Suppression also replaces values of an attribute with a special key (Bayardo and Agrawal 2005; Aggarwal et al. 2005), typically an asterisk (*). As data are suppressed, identifiable values are replaced by special keys to make values non-identifiable. Suppression is generally applied to explicit-identifiers and quasi-identifiers. Suppression ensures personal information is not disclosed.

On the other hand, Perturbation performs anonymization by distorting the original data with the addition of noise, data swapping, value aggregation and generation of synthetic data. Statistical approaches are used to perturb data values (Aggarwal et al. 2005). Perturbation generally replaces real data values as well so that data does not correspond at all to the original value associated with the individual. When statistical methods are used to perturb data, while attribute values are not those of the original individual, the aggregate characteristics of the attributes are the same as for the entire dataset.

### General Anonymization Techniques

Anonymization Techniques use combinations of the Anonymization Operations described above to anonymize PII. The most popular techniques used to limit disclosure of identifiable information are K-anonymity-based techniques and Differential Privacy. The anonymization techniques address privacy protection under different circumstances of access to data.

#### K-Anonymity-Based Techniques

K-anonymity-based techniques are relevant in the following data access circumstances. The original dataset is contained in one or multiple tables and all Explicit Identifiers have been removed. K-anonymity requires that after removal of Explicit Identifiers, each record must be indistinguishable from at least another \( k - 1 \) records with respect to any given quasi-identifier (Sweeney 2002; Aggarwal et al. 2005). For example, when \( k \)-anonymized, if a given record has a given value for an attribute, there will be at least \( k - 1 \) other records with that same value. As such, \( k \)-anonymization removes the uniqueness of distinct values for a quasi-identifier through generalization and suppression operations.

While \( k \)-anonymity protects against identity disclosure, it is insufficient to prevent attribute disclosure (being able to associate a unique attribute value to a given record). L-diversity on the other hand is a concerned not so much with identity disclosure, but with the ability to associate Sensitive Attributes to a given record Machanavajjhala et al. (2006). An equivalence class (i.e., a set of records that are indistinguishable from each other with respect to a given quasi-identifying attribute) is said to have l-diversity if there are at least \( l \) “well-represented” values for the sensitive attribute. As such, this is fundamentally \( k \)-anonymity but for the special case of Sensitive Attributes (Machanavajjhala et al. 2006). A table is said to have l-diversity if every equivalence class of the table has l-diversity. As a result, and like \( k \)-anonymization, l-diversity removes the uniqueness of...
distinct values, but for a sensitive attribute through generalization and suppression operations. A common problem of both $k$-anonymity and $l$-diversity is that they cannot guarantee the protection of private data if information about the global distribution of an attribute is known, e.g., if someone had access to the entire table containing any given $k$-anonymized or $l$-diversified attribute. This problem is particularly acute if the distribution of the attribute in question has few values and/or is highly skewed towards a few values, e.g., if 90% of the values of tips given to drivers (see example in “Challenges and Opportunities in Privacy Protection”) in a given dataset were 0, it would be straightforward to infer that a given individual did not leave tips. To address this problem, the $t$-closeness anonymization technique has been developed.

$t$-Closeness (Li et al. 2007) itself is a measure of the degree to which a distribution is skewed towards a few values. As $t$-closeness increases, a distribution becomes more skewed towards a few variables. The $t$-closeness technique amounts to adjusting the distribution of sensitive attributes to assure that the global distribution does not have few values and is not highly skewed towards any, or a few, of those values. An equivalence class is said to have $t$-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a given threshold $t$. A table is said to have $t$-closeness if all equivalence classes have $t$-closeness. $t$-closeness is ensured through generalization and suppression operations.

**Differential Privacy**

Strictly speaking, Differential Privacy is not a technique, but rather a property of the anonymization process. The concept of Differential Privacy was originally introduced by Dwork et al. (2008) and is relevant in the following data access circumstances. There is an original database ($D$) with Explicit- and/or Quasi-identifiers and Sensitive Attributes. There are also two agents accessing the data either indirectly or directly. The data user wants to learn about the characteristics of the original dataset by making queries to it, but does not have direct access to the original data. The Curator (a software component between other software layers, or middleware) has direct access to the original data, but has the role of modifying it thus creating a new dataset ($D'$) to which the user has direct access.

Critical to understanding Differential Privacy is the notion of Privacy Degradation. Privacy Degradation describes the fact that as queries are made to a database, the results of each additional query provide information that can be compared with previous results. As such, it is possible, all else equal, to learn about individual observations in a modified database by comparing results made with different queries.

Ultimately, the Curator’s role is twofold. First, remove Explicit identifiers from the original database and perform modifications (Perturbations) on the quasi-identifiers and Sensitive Attributes. These perturbations are typically created by adding noise drawn from a Laplace distribution to quasi-identifiers and Sensitive Attributes. It is important to note that $D$ itself is dynamic, so that it might not be the same for subsequent queries from the user. The degree to which $D$ is different from $D'$ is referred to as epsilon ($\varepsilon$). With Differential Privacy $\varepsilon$ is also dynamic and is a function of the number of queries from the user.

**Location Privacy**

Anonymization methods discussed so far have been developed and applied primarily to PII (Schwartz and Solove 2011; McCallister et al. 2010) data. The large amounts of temporally and spatially precise BTD can be thought of as quasi-identifying data, but the techniques mentioned above are not suitable to ensure privacy protection with this PILI data. There are two broad categories of circumstances under which anonymization of PILI can take place. The first relates to when data are transferred from the data recorder to the data collector. The second is when data are transferred from the data collector to the data analyst. The first category is referred to as location-based query (LBQ) anonymization (Kalnis et al. 2007; Ghinita et al. 2008). This might happen, for example if the true location of the data owner is anonymized or obscured by the data recorder before being sent as part of an LBQ, such as a search for nearby restaurants. The second category is when data are transferred from the data collector to the data analyst. While the first type of anonymization is important, we believe it to be less of a challenge to the use of BTD than the second. This is because with LBQs the data collector and Analyst will typically be known and presumably trusted if the data owner is willing to share their information with them. Of greater practical concern is what happens as data are transferred from the collector to the analyst, since the identity of analysts may not be known, and there may be many, particularly in the case of open data. As a result, in this paper we concentrate on techniques relevant for anonymization that takes place between the data collector and the data analyst, i.e., to data “publishing”.

There are four main techniques available for location anonymization appropriate for PILI data when it is published. The techniques differ along three dimensions: whether a data owner ID persists or not; the approach used for obfuscating location; and whether or not the anonymization is done real time.
Spatial Cloaking

Spatial cloaking (Gruteser and Grunwald 2003; Chow et al. 2006) is used when data are static (i.e., when it has already been recorded and stored). When location data are spatially cloaked the data owner IDs persist across observations, but locations reported to the data user are adjusted. In particular, and instead of providing the original location data (i.e., latitude and longitude), the data are spatially aggregated so that the data user is provided a spatial buffer known as the anonymized spatial region (ASR) (Terrovitis and Mamoulis 2008). The size of the buffer is dynamic and is a function of the number of other data owners on whom data are reported. In particular, the ASR is large enough to encompass the data of at least k-other data owners. As such, it can be seen as a spatial k-anonymization. Since ASRs are dynamic this technique is also computationally intensive. This technique could be used with trip-end location data or trajectory data.

Mixed Zones

Mixed zone (MZ) (Beresford and Stajano 2004) anonymization is used when data are static. With MZ-anonymization, it is data owner (pseudonym) IDs that are obfuscated and not their locations. This is done first by defining zones through which the data owner passes. As the data owner passes through zones, their IDs are modified so that it is not possible to follow an individual as they pass through the different zones. Mixed zones is a more general approach that encompasses the special case of the vehicular mix zone approach.

Dummy Trajectories

As with spatial cloaking and mixed zones, the dummy trajectories technique (You et al. 2007) is used to anonymize static spatial data, particularly static trajectory data. As with spatial cloaking, data owner (pseudonym) IDs persist throughout the data and unlike MZ does not involve the creation of zones. This method amounts to perturbing location data over the course of a trajectory.

Path Confusion

As with spatial cloaking and dummy trajectories with path confusion (Hoh and Gruteser 2005) data owner IDs persist. Like the dummy trajectory approach, the location data are perturbed directly. Unlike these other methods, the data in question is not static but is arriving in real time. The key concern with this approach is to make it impossible to predict a future location based on the dynamic data. As such, this is a more statistically involved approach that not only perturbs location data directly, but also associated bearing and speed data. Due to the statistical complexity and the need to treat each data point in real time, it is computationally intensive.

Cross-Cutting Opportunities and Challenges

The previous sections have focused on the primary challenges facing the widespread use of BTD and the opportunities to overcome these challenges. The opportunities in these sections have included those that are applicable to one of the challenges at a time. In this section, we discuss opportunities (and challenges associated with their implementation) that will help in overcoming more than one of the “3-V” challenges. In one way or another the approaches required to overcome the 3-V challenges amount to being able to add computing resources, constrained ultimately by hardware. Cloud computing (Armbrust et al. 2010) involves adding resources virtually. That is to say that instead of adding physical resources (e.g., servers), it is possible to add resources through software that mimic the behavior of physical hardware. This can be done “privately” on infrastructure managed directly or “publicly” by going through Cloud computing providers such as Amazon Web Services (AWS), Google Cloud, Microsoft Azure, Rackspace, etc. Cloud computing allows the possibility to quickly add resources, and thereby scale systems in near real time and even automatically. Using Cloud resources reduces the requirements for internal expertise and allows granular addition of resources where the addition of physical infrastructure is “lumpier”.

The costs of using Cloud computing services need to be traded-off with the costs of managing physical infrastructure, but is becoming increasingly competitive for almost all typical computing requirements. It is likely to become even more competitive over time making the choice of using Cloud computing somewhat easier on a cost-only basis. Another issue with Cloud computing, however, is the loss of control over where data are physically stored (i.e., where physical servers are located). This can be an issue for transportation authorities that have traditionally operated under circumstances where all data are stored “internally”. Of course, Cloud computing can be done “privately” although this still requires a great deal of internal resources (more than managing infrastructure directly) and is likely only viable for large organizations.

Cloud resources (Bhardwaj et al. 2010) can be added through three service models: infrastructure as a service (IaaS), software as a service (SAAS) and platform as a service (PAAS). IaaS is the most direct model for adding additional resources. It involves the addition of virtual infrastructure (e.g., computers) that are managed by the service user. As such, software required by the user is installed and maintained on the additional virtual resources. SAAS is the most limited model with users subscribing to particular
application software and databases. Microsoft Office online, SQL Server web, ArcGIS online are all examples of this. PAAS is the most involved of the three models. PAAS solutions are designed primarily for technology developers and as a result provide all necessary elements of a development environment. That is PAAS comes with pre-packaged operating system, web server, database and programming languages. PAAS examples include IBM Cloud, Microsoft Azure, Blockchain (Lopez and Farooq 2018), etc.

The Future of BTD in Transportation

The rapid emergence of different tools for data collection has led to an unprecedented potential not only to collect, but to integrate data from many sources and potentially to revolutionize how transportation planning and operations are done. This potential has not been lost to transportation researchers, but current research has focused on techniques for collecting data or on inferring relevant transportation information from this data. While critical to fulfilling the potential, we define four existing, higher-level challenges and opportunities to the large-scale use and integration of BTD for planning and decision-making purposes. Three of the challenges (and opportunities) are related directly to the 3-Vs of Big Data more generally. A fourth relates to all of the 3-V challenges collectively, and a fifth concentrates on the challenge of privacy protection. This is particularly relevant to BTD due to the large amount of temporally and spatially precise data collected. In our view, BTD will not be able to fulfill its promise if these challenges are not met. We consider the challenges related to the 3-Vs (volume, variety and velocity) first and continue with those related to privacy.

Challenges Associated with the 3-Vs

The challenge associated with the sheer volume of BTD that are, and that will continue to be, available in ever-larger quantities will continue to place pressure on traditional vertically based database management systems. The ability to vertically scale these systems is already at its limits and as a result the future will increasingly (and perhaps eventually entirely) require horizontally scaled systems deployed using distributed architectural approaches. While current approaches are dominated by CA architecture, there is a gradual drift towards AP architecture ensuring high availability, fault tolerance and delayed or eventual consistency. This pattern will continue into the future and AP architectures are likely to become the dominant approach in the near, and for the foreseeable future.

While large volumes of data present their own challenge, being able to process data coming in at different rates and increasingly in real time is the challenge of velocity. Traditional batch processing methods are ill-adapted to the onslaught of real-time data that also needs to be processed in real time. As a result, in the future the need to increasingly devote resources to stream processing methods will become more prominent. While stream processing will undoubtedly make up a larger proportion of processing, batch processing, when appropriate will continue to play an important role. Batch processing will remain the mainstay of processing for static datasets and analysis requiring access to a finite dataset. In the future, processing will not simply take place as batch or stream processing, but is likely to involve techniques that take advantage of both approaches, such as emerging “Lambda” architectures (Serra 2018).

The variety (different data and file formats from different sources) of BTD represents another key challenge. Traditional structured relational database management systems that require defined data schemas are incapable of handling and integrating data from different sources; something necessary but also which provides one of the most important aspects of the potential of BTD. As such, the move away from traditional RDBMSs and towards more flexible non-relational DB systems will need to continue to cope with the many different formats. The most common production-ready flexible systems are NoSQL-based and such systems are set to become more commonplace and the de facto standard in the near future. At the same time, new approaches are already evolving to overcome the constraints of NoSQL systems and in particular new flexible systems that are also ACID compliant with NewSQL-systems being the most likely to replace NoSQL.

Finally, Cloud computing will be key to meeting all of the 3V challenges. It will provide the possibility to granularly, quickly and automatically add computing resources necessary to cope with increased volumes, velocity and variety of data. The economic case for Cloud computing seems undeniable, but its use will likely involve the necessity to give up the ability to store data internally for most organizations. As such, in order for it to play its facilitating role in allowing BTD to revolutionize planning, organizations will need to be convinced that collected data are stored sufficiently safely. This will likely happen through a combination of attempts on the part of Cloud service providers to convince organizations of the safety of data and an eventual institutional acceptance of using these services.

Challenges Associated with Cyber-Security and Privacy

The last major challenge is that of security and privacy. The first three challenges are essentially technical in nature and if not met, it will simply not be possible to take advantage of the potential of BTD. Privacy on the other had is both a technical, as well as social/political challenge. The social/
political challenge is that of data owners (the public) being willing to share their data with data collectors and subsequently to data users. Ensuring this willingness has three elements. The first is that related to security, which is a challenge facing all IT. Network threats have not been dominant in the transport industry as compared to other sectors. Notwithstanding, there is a rising need to build robust and secured transportation infrastructure that is protected from wide range of system vulnerabilities and exploits. As a step to improve security, network threats to existing systems need to be assessed and reported. Network assessment tools [e.g., Wireshark (Orebaugh et al. 2006)] have become popular for network monitoring to gain better visibility of vulnerability to cyber threats. Enterprise architectures deploy network security systems such as firewalls and proxy servers, which monitors incoming and outgoing network traffic by adhering to strict security rules. A final step to achieve secured computing is to improve communication between trusted cyber-security experts and operators at national and local transportation agencies. Such communication notifies on active security threats and provides relevant information on managing such threats.

The second relates to the knowledge of data users with respect to the nature of their data that are being shared as well as with whom. This has been prominent lately with the necessity for companies to comply with European GDPR regulations. We believe an important challenge related to this is also in the simple and clear explanation to the data owners of what data are being collected and shared, something not easily accessible through typical terms of use and consent forms. The third is that related to privacy protection and more specifically privacy protection in the context of “published” data. That is, data that are shared with data users. Traditionally data were published to relatively few people whose identities were known. The advent of open data has resulted (and will increasingly result in the future) in many more people whose identities are not known having access to published data. Moreover, with the anonymity of data users, the risk of the adversarial use of such data, particularly with increasing “background” knowledge, and therefore the threat of privacy breaches will only increase. As such, ensuring willingness on the part of data owners will increasingly involve assurances around the protection of privacy from collected data. These assurances will be based upon methods of anonymization. As a result, anonymization is critical to ensure the trust of data users.

There are already many anonymization techniques that have been developed for the purposes of privacy protection with both tabular as well as geographic data, and this is a lively area of academic and private sector research. At the same time, this is an evolving field and one that will have to continue to evolve as more data becomes open. The primary reason for this is the growth of open data for two reasons. First, as more data become open, there will be more people able to access it anonymously and as a result a greater threat of adversarial use of the data. Compounding is the fact that as more data becomes open, more “background” knowledge will also become open further expanding the threat of potential privacy breaches. As a result, not only will it be necessary for anonymization techniques to evolve, but caution related to the data that are made open will need to be taken.

Funding Funding was provided by Social Sciences and Humanities Research Council.

References

Abadi D (2016) Optimizing disk io and memory for big data vector analysis. http://blogs.teradata.com/data-points/optimizing-disk-io-and-memory-for-big-data-vector-analysis/. Accessed 17 Aug 2018

Aggarwal G, Feder T, Ksentini A, Motwani R, Panigrahy R, Thomas D, Zhu A (2005) Anonymizing tables. In: International conference on database theory. Springer, 2005, pp 246–258

Amiri S, Gerostathopoulos I, Prehofer C (2017) Big data analytics architecture for real-time traffic control. In: Models and technologies for intelligent transportation systems (MT-ITS), 2017 5th IEEE international conference on, IEEE, 2017, pp 710–715

Anderson JC, Lehndt J, Sater N (2010) CouchDB: the definitive guide: time to relax. O’Reilly Media Inc, Newton

Arentze T, Timmermans H, Hofman F, Kalis N (2000) Data needs, data collection, and data quality requirements of activity-based transport demand models. Transp Res C (E-C008), p 30

Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I et al (2010) A view of cloud computing. Commun ACM 53(4):50–58

Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z (2007) Dbpedia: a nucleus for a web of open data. In: The semantic web. Springer, pp 722–735

Bagchi M, White PR (2005) The potential of public transport smart card data. Transp Policy 12(5):464–474

Barcelo J, Montero L, Marques L, Carmona C (2010) Travel time forecasting and dynamic origin-destination estimation for freeways based on bluetooth traffic monitoring. Transp Res Rec J Transp Res Board 2175:19–27

Bayardo RJ, Agrawal R (2005) Data privacy through optimal k-anonymization. In: Data engineering, 2005. ICDE 2005. Proceedings. 21st international conference on, IEEE, 2005, pp 217–228

Beresford AR, Stajano F (2004) Mix zones: user privacy in location-aware services. In: Pervasive computing and communications workshops, 2004. Proceedings of the second IEEE annual conference on, IEEE, 2004, pp 127–131

Bhardwaj S, Jain L, Jain S (2010) Cloud computing: a study of infrastructure as a service (iaas). Int J Eng Inf Technol 2(1):60–63

Bierlaire M, Chen J, Newman J (2013) A probabilistic map matching method for smartphone GPS data. Transp Res Part C Emerg Technol 26:78–98

Bohte W, Maat K (2009) Deriving and validating trip purposes and travel modes for multi-day GPS-based travel surveys: a large-scale application in the netherlands. Transp Res Part C Emerg Technol 17(3):285–297
Hainen A, Wasson J, Hubbard S, Remias S, Farnsworth G, Bullock D (2011) Estimating route choice and travel time reliability with field observations of Bluetooth probe vehicles. Transp Res Rec J Transp Res Board 2256:43–50
Hasan O, Brunie L, Bertino E, Shang N (2013) A decentralized privacy preserving reputation protocol for the malicious adversarial model. IEEE Trans Inf Forensics Secur 8(6):949–962
Hashem IAT, Yaqoob I, Anuar NB, Mohktar S, Gani A, Khan SU (2015) The rise of big data on cloud computing: review and open research issues. Inf Syst J 47:98–115
Hilbert M, Lopez P (2011) The world’s technological capacity to store, communicate, and compute information. Science 332(6025):60–65
Hoh B, Gruteser M (2005) Protecting location privacy through path confusion. In: Security and privacy for emerging areas in communications networks. 2005. SecureComm 2005. First international conference on, IEEE, 2005, pp 194–205
Hood J, Sall E, Charlton B (2011) A GPS-based bicycle route choice model for San Francisco, California. Transp Lett 3(1):63–75
Jagadish H, Gehrke J, Labrinidis A, Papakonstantinou Y, Patel JM, Ramakrishnan R, Shahabi C (2014) Big data and its technical challenges. Commun ACM 57(7):86–94
Ji C, Li Y, Qiu W, Awada U, Li K (2012) Big data processing in cloud computing environments. In: Pervasive systems, algorithms and networks (ISPAN), 2012 12th international symposium on. IEEE, 2012, pp 17–23
Kahn SD (2011) On the future of genomic data. Science 331(6018):728–729
Kalnis P, Ghinita G, Mouratidis K, Papadias D (2007) Preventing k-anonymity and l-diversity: privacy beyond k-anonymity. In: Data engineering, 2007. ICDE’07. Proceedings of the 22nd international conference on, IEEE, 2007, pp 52–61
Kish LB (2002) End of Moore’s law: thermal (noise) death of transistor models. Phys Lett A 290(3–4):144–149
Krzanich B (2016) Data is the new oil in the future of automated driving. https://newsroom.intel.com/editorials/krzanich-the-future-of-automated-driving/. Accessed 13 Aug 2018
Lagoze C (2014) Big data, data integrity, and the fracturing of the control zone. Big Data Soc 1(2):205951714558281
Lakshman A, Malik P (2010) Cassandra: a decentralized structured storage system. ACM SIGOPS Oper Syst Rev 44(2):35–40
Leduc G (2008) Road traffic data: collection methods and applications, 2008. Springer, 2010, pp 25–36
Li X, Li Y, Li H, Li X, Shen H, Yuan J, Fan Y (2015) A framework for spatio-temporal big data scalable stream computing platform. In: Data mining workshops (ICDMW), 2010 IEEE international conference on, IEEE, 2010, pp 170–177
Lindell Y (2005) Secure multiparty computation for privacy preserving data mining. In: Encyclopedia of data warehousing and mining, IGI global, 2005, pp 1005–1009
Lopez D, Farooq B (2018) A blockchain framework for smart mobility, submitted to the Blockchain technology symposium (BTS’18)—from hype to reality. The Fields Institute, Toronto (September, 2018)
Lv Y, Duan Y, Kang W, Li Z, Wang F-Y (2015) Traffic flow prediction with big data: a deep learning approach. IEEE Trans Intell Transp Syst 16(2):865–873
Machanavajjhala A, Gehrke J, Kifer D, Venkitasubramaniam M (2006) l-diversity: privacy beyond k-anonymity. In: Data engineering, 2006. ICDE’06. Proceedings of the 22nd international conference on, IEEE, 2006, pp 24–24
Maier D (1983) The theory of relational databases, vol 11. Computer Science Press, Rockville
Mansuri IR, Sarawagi S (2006) Integrating unstructured data into relational databases. In: Data engineering, 2006. ICDE’06. Proceedings of the 22nd international conference on, IEEE, 2006, pp 29–29
Marz N (2013) Storm: Distributed and fault-tolerant realtime computation. https://www.infoq.com/presentations/Storm-Introduction
MccAllister E, Grance T, Scarfone KA (2010) Sp 800-122, guide to protecting the confidentiality of personally identifiable information (pii)
McGowen PT, McNally MG (2007) Evaluating the potential to predict activity types from GPS and GIS data. In: Proceedings of annual meeting of the transportation research board, transportation research board, Washington, DC, 2007, reference number: 07-3199
Mikkelsen MR, Christensen P (2009) Is children’s independent mobility really independent? A study of children’s mobility combining ethnography and GPS/mobile phone technologies. Mobilities 4(1):37–58
Moniruzzaman ABM, Hossain SA (2013) Nosql database: New era of databases for big data analytics-classification, characteristics and comparison, arXiv:1307.0191
Montini L, Prost S, Schrammel J, Rieser-Schussler N, Axhausen KW (2015) Comparison of travel diaries generated from smartphone data and dedicated GPS devices. Transp Res Procedia 11:227–241
Nergiz ME, Atzori M, Saygin Y (2008) Towards trajectory anonymization: a generalization-based approach. In: Proceedings of the SIGSPATIAL ACM GIS 2008 international workshop on security and privacy in GIS and LBS, ACM, 2008, pp 52–61
Neumeyer L, Robbins B, Nair A, Kesari A (2010) S4: distributed stream computing platform. In: Data mining workshops (ICDMW), 2010 IEEE international conference on, IEEE, 2010, pp 170–177
Orebaugh A, Ramirez G, Beale J (2006) Wireshark & ethereal network protocol analyzer toolkit. Elsevier, Amsterdam
Orru M, Paolillo R, Detti A, Rossi G, Melazzi NB (2017) Demonstration of opengeopecake: the ICN nosql spatio-temporal database. In: Local and metropolitan area networks (LANMAN), 2017 IEEE international symposium on, IEEE, 2017, pp 1–2
Ousterhout J, Douglis F (1989) Beating the i/o bottleneck: a case for log-structured file systems. ACM SIGOPS Oper Syst Rev 23(1):11–28
Xu L, Jiang C, Wang J, Yuan J, Ren Y (2014) Information security in big data: privacy and data mining. IEEE Access 2:1149–1176
Yazdizadeh A, Patterson Z, Farooq B (2019) An automated approach from GPS traces to complete trip information. Int J Transp Sci Technol 8(1):82–100
You TH, Peng WC, Lee WC (2007) Protecting moving trajectories with dummies. In: Mobile data management, 2007 international conference on, IEEE, 2007, pp 278–282
Zahabi SAH, Ajzachi A, Patterson Z (2017) Transit trip itinerary inference with GTFS and smartphone data. Transp Res Rec J Transp Res Board 2652:59–69
Zhang J, You S, Gruenwald L (2014) High-performance spatial query processing on big taxi trip data using gpgpus. In: Big data (Big-Data Congress), 2014 IEEE international congress on, IEEE, 2014, pp 72–79
Zhao F, Ghorpade A, Pereira FC, Zegras C, Ben-Akiva M (2015) Stop detection in smartphone-based travel surveys. Transp Res Procedia 11:218–226
Zheng X, Chen W, Wang P, Shen D, Chen S, Wang X, Zhang Q, Yang L (2016) Big data for social transportation. IEEE Trans Intell Transp Syst 17(3):620–630
Zikopoulos P, Eaton C et al (2011) Understanding big data: analytics for enterprise class hadoop and streaming data. McGraw-Hill Osborne Media, New York

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.