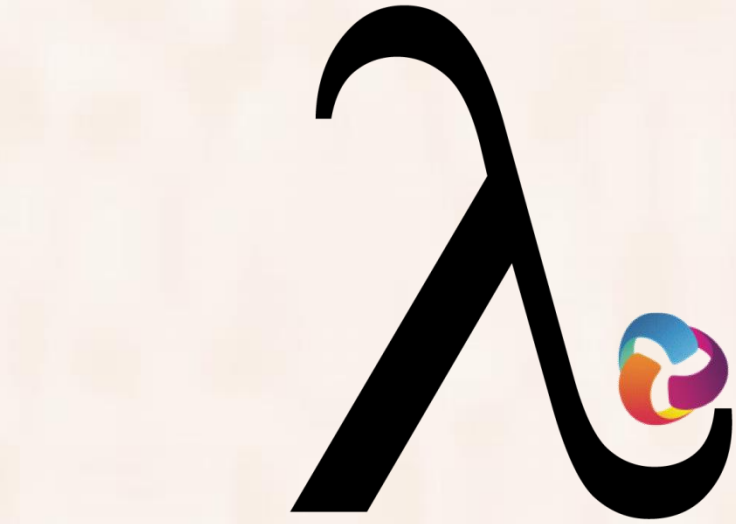

LEARNING, APPLYING, MULTIPLYING BIG DATA ANALYTICS

Horizon 2020 Grant Agreement No 809965
Contract start date: July 1st 2018, Duration: 30 months

LAMBDA Deliverable 2.1
Big Data Challenges and Analysis of Scientific and Technological Landscape

Due date of deliverable: 31/12/2018
Actual submission date: 28/12/2018

Revision: Version 1.0

Dissemination Level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme, H2020-WIDESPREAD-2016-2017 Spreading Excellence and Widening Participation under grant agreement No 809965.

Author(s)	Dea Pujić (PUPIN), Marko Jelić (PUPIN), Hajira Jabeen (UBO)
Contributor(s)	
Internal Reviewer(s)	Tim Furche (UOXF)
Approval Date	
Remarks	

Workpackage	WP 2 Exploiting Synergies and Setting the Twinning Initiative
Responsible for WP	Fraunhofer Institute for Intelligent Analysis and Information Systems
Deliverable Lead	Institute Mihajlo Pupin (Valentina Janev)
Related Tasks	Task 2.1 Science and Technology watch

Document History and Contributions

Version	Date	Author(s)	Description
0.1	10.12.2018	Dea Pujić, Marko Jelić	Contribution
0.2	20.12.2018	Hajira Jabeen	Contribution
0.3	21.12.2018	Tim Furche	Review
0.4			

Executive Summary

This deliverable entitled “Big Data Challenges and Analysis of Scientific and Technological Landscape” gives an overview of the Big Data concepts, outlines some of the relevant challenges in this domain and reviews and describes the current state of the art tools relevant to Big Data applications. With the potential use cases for Big Data in mind, particular technological barriers are imposed and specialized software solutions, frameworks and libraries, described in this document, need to be employed in order to surpass the limitations of traditional methods for data storage and analysis.

Table of contents

Abbreviations and Acronyms

BDA	Distributed Big Data Analytics
ARCH	Big Data Architecture
R&D	Research and Development
R&D&I	Research and Development and Innovation
SSE	South-East Europe Countries
IoT	Internet of Things
SemTech	Semantic Technologies
SWOT	Strengths, Weaknesses, Opportunities and Threats.
VIS	Visualization

List of Figures

List of Tables

1. Introduction

With one of the goals of the LAMBDA project (**L**earning, **A**pplying, **M**ultiplying **B**ig **D**ata **A**nalytics)[1] being the application and transfer of knowledge between institutions and also advancing the general degree of education with the researchers and other staff of Institute Mihajlo Pupin[2] which would later be projected onto the neighbouring region and network of academic institutions in the field of Big Data research and analysis, this report aims to summarize the current state of the art solutions of the Big Data landscape.

In this deliverable, contemporary technological solutions from relevant research papers and articles as well as frameworks and concepts that are applied in industrial applications are listed, described, analysed and compared. With the amount of data being generated and processed growing on a daily basis, the need for newer and faster answers to new technological barriers is rapidly increasing and so the landscape of Big Data solutions is actively changing and evolving. This report is set to capture the Big Data scenery in its current shape.

The remainder of this report is organized as follows: Section 2 elaborated on the background of Big Data, gives its general definition, a brief historical overview and some attributes that are often used when describing what Big Data is. Challenges regarding the development of Big Data solutions are also listed as well as both general applications of Big Data and those relevant for Institute Mihajlo Pupin. Moving on, Section 3 characterizes the current landscape of Big Data by analysing the most popular frameworks used to handle the required amounts of data, gives an overview of database management solutions stemming from some of the challenges posed by Big Data and outlines how knowledge can be extracted from those data stores by using graphs and relational structures, lists and describes stream processing engines, outlines pre-processing required for proper analytics on Big Data as well as explaining how analytics can be performed. Finally, Section 3 closes with comparing different software solutions for data visualization. At the end, Section 4 finalizes the discussion with general remarks about the Big Data landscape.

[1] http://www.project-lambda.org/
[2] http://www.pupin.rs/

2. Background

2.1 Big Data Definition and History

Big Data has recently become more used as a buzzword than a precisely determined scientific object or phenomena. Having a wide variety of uses and many different sorts of applications and implementations further increases the difficulty of pinning down what Big Data actually is. Nevertheless, even though Big Data has no formal single definition, in general it can refer to volumes of data that cannot be (efficiently) processed by conventional data management methods and storage systems or, more precisely, it is sometimes used to describe large datasets with sizes in the range of exabytes (1e18 bytes) and greater. Such large streams of data are becoming more and more common with analogue technologies slowly being phased out and replaced with digital solutions that can hugely benefit from storing recorded data, performing analysis on that data and information (often also referred to as knowledge) being inferred from it. Processing such high volumes poses a challenge even for the modern day IT infrastructure and establishment and so innovative architectures and handling methods needed to be developed to facilitate the heavy process of Big Data management.

The history of Big Data could be linked back to the 20th century or more precisely to 1937 when social security was introduced in the United States of America and the requirement arose for data management of 26 million residents. This job was handed to IBM who developed a punch card machines for reading data. Around twenty years later, during World War II, a quick (for that day and age) pattern recognition system was used by the British to extract intelligence from intercepted messages at a rate of five thousand characters per second. Data analytics was continued during the Cold War with the American National Security Agency hiring around twelve thousand cryptologists to facilitate the knowledge discovery demand. The first data center in the world was planned in 1965 to store tax returns and fingerprint records which were meant to be later converted to magnetic tapes but this project was later scrapped due to the controversies that it generated. The invention of the World Wide Web in 1989 must also be mentioned as a significant date as it is the source of data with the most presence today and is used as the main means of data exchange. A few years later, supercomputer concepts started to emerge, with promising processing power outlooks. With the start of the 21st century, promising new technologies started to arise like the Google File System in 2003 (Ghemawat, Gobioff, & Leung, 2003) and the introduction of MapReduce in 2004 (Dean & Ghemawat, 2014) that is still being used today for distributed processing. 2005 and 2006 saw the start of development on Hadoop which came into production in 2008 (Shvachko, Kuang, & Radia, 2010). Following this, Yahoo released Pig (Olston, Reed, Srivastava, Kumar, & Tomkins, 2008) and Facebook created Hive in 2009, with S4 (Chauhan, Chowdhury, & Makaroff, 2012) and Flume released in the following year. 2011 marked the start of real time as opposed to batch processing with Storm (Surshanov, 2015) and Spark (Zaharia, Chowdhury, Franklin, Shenker, & Stoica, 2010) being released that year. 2012 saw the creation of Kafka by LinkedIn (Kreps, 2011) and the definition of the Lambda architecture for efficient Big Data processing with 2013 giving as Samza by LinkedIn and Flink (Friedman & Tzoumas, 2016) and the Kappa architecture being defined in 2014. This year also marked the beginning of hybrid data processing with 2015 giving us Cloudera Kudu and Google's Beam. The Big Data paradigm has definitely caught on with many research projects funded by various institutions already finished (Big Data Europe, SANSA, etc.) and even more currently ongoing with promising new results like the Vadalog System for Knowledge Graphs developed by the University of Oxford within the VADA (Value Added Data Systems) research programme (Bellomarini, Gottlob, & Sallinger, 2018).

When the concept of Big Data was originally introduced, its main characteristics were depicted using the so-called 3V's of Big Data being (Doug, 2011), (Lakshen, Vraneš, & Janev, 2016):

- **Volume** - the sheer amount of data being created from various sources like medical data, research studies, web pages, text documents, audio and video media, meteorological data,

government records and so on. For example, statistical data states that several hundred hours of video material are being uploaded to YouTube every minute. With region-dependant regulatory standards banning certain types of content, this data must be somehow processed. However, human processing is not possible at this scale due to the high volume of data, and therefore sophisticated solutions like machine learning must be employed. Similar problems also exist with other user-generated digital content like photographs with over a trillion taken yearly.

- **Velocity** - the rate at which data is being created, refreshed or renewed, resulting in high volumes mentioned previously. Examples of this behaviour can be found in social network posts, search engine queries and so on.
- **Variety** - the differences in structure of the data input to the system. Architectures based on big data have to be capable of processing heterogeneous data that can be both structured and unstructured. Some architectures were even developed to handle semi-structured data as an intermediary solution. Obviously, data encoded in mark-up languages, database dumps, sensor data, documents, photographs, videos and audio data are all differently formatted but must be processed by a unified system somehow.

More recent studies have identified more V's forming up to 5V's of Big Data with the addition of (Kalbandi & Anuradha, 2015):

- **Veracity** - the truthfulness or uncertainty of the data. Modern day data processing tools can generally benefit from having vast amounts of data like for example in machine learning training algorithms. However, this data must be trusted in order to be used and for the derived result to be considered valid. Knowing and trusting the source of the data collection, the assurance that no data classes are underrepresented or that if they are that is properly documented, that all the data is collected in equal conditions, that the correct methodology was in place when collecting data and similar questions should be posed in order to eliminate veracity problems that may plague the data.
- **Value** - the end result, or simply information derived from raw Big Data. This is arguably one of the most important V's from this list as it is considered the goal of any alike data processing techniques especially having in mind that using Big Data involves significant technological and infrastructural improvements over traditional simple data processing structures. Whether it be pattern recognition, customer behaviour analysis, product placement, accelerating business, credit or insurance rankings, being able to find value in Big Data is considered the key for even considering the possibility of using solutions based on Big Data. Since Big Data is generally employed by business on large scales, the value extracted must surpass the investment and maintenance costs for it to be a viable option. In order to achieve this, efficient storage solutions, analytical programs and consumer friendly result displaying applications must be used.

Moreover, this list can be supplemented with two additional V's further extending the characterization of Big Data with 7V's, as depicted in Figure 1 (Khan, Uddin, & Gupta, 2014):

- **Validity** - how suitable the selected data is for the application at hand. Although validity is very similar to veracity, with the two often getting mixed up and used interchangeably, validity only considers if the selected dataset that is collected or processed is accurate and correct in the context of intended usage. Examples in literature mention a case where a doctor has to make a decision about patient treatment based on clinical trials, but poses a question whether he should trust the raw data before validating it. Alike problems should be of no surprise given the fact that most of the data scientist's time is usually occupied by cleaning and filtering the data for the specific use cases that are to be examined.
- **Volatility** - the temporal validity of the data. A volatile set of data is a set which can be considered no longer significant, irrelevant or historic after a set amount of time has passed. Although in the sphere of traditional database solutions, "never delete data" was a guideline that could often be heard, a question should be posed whether and after how long can a

piece of stored data be ignored or removed. With various new volumes of data being created with high velocity in every second, having a constant stream of relevant data replacing outdated old entries is crucial to maintaining the system.

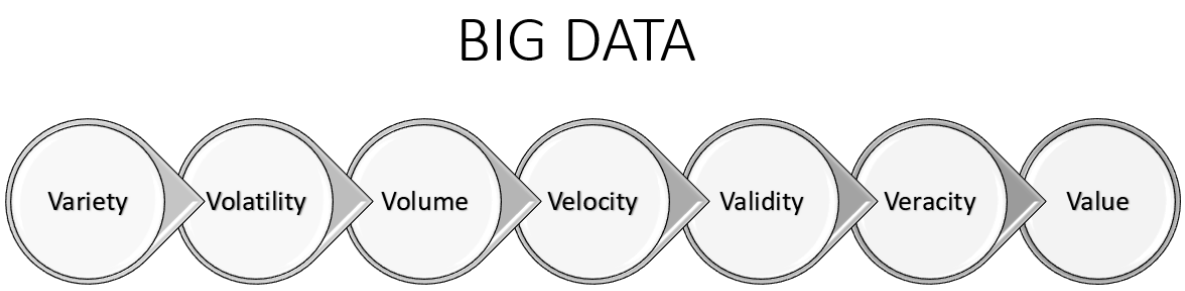

Figure 1. The seven V's of Big Data

Finally, related literature (Khan, et al., 2018) also suggests expanding the mentioned 7V's to 10V's by adding

- **Visualization** - how to display the gathered information obtained from Big Data processing. Traditional means of showcasing information are of no use in this regard because of the large amount and diversity associated with Big Data that reach over the limits of in-memory technologies used in established applications. Two main problems are attached with Big Data visualization: How to adequately represent the data in a user-friendly way and what resources and technologies are necessary. Render times and scalability must also be considered when going beyond the familiar depiction of data by using 2D graphs, charts and so on. The software tools used in this regard range from pure scientific statistical computing platforms like Matlab, Octave, Python, R that deal well with increasing volumes of data and are good for raw data visualization to Business oriented solutions like Tableau or JupytR that tackle high variety and inherently more informative than just displaying data (Caldarola & Rinaldi, 2017).
- **Vulnerability** - the security issues attached to Big Data. Just possessing the huge amounts of data required for running Big Data analytical solutions puts a business at a high risk because it must uphold the privacy standards and protect the data from leaks, theft and misuse. With the contemporary paradigm of software users actually being products meaning that the information gathered by the software during use is later applied for optimizing advertising engines, giving better predictions or outright sold in an anonymized form, all of our personal information that we easily put up online may be at risk from hackers and thieves. Even though the major software companies state that all information is kept confidential and/or encrypted, database attacks and unauthorised data access are slowly becoming commonplace leading to personal information being sold on the dark web. When discussing this topic, the "Cambridge Analytica data scandal" comes to mind where Cambridge Analytica had collected personal information that would later be used for political agendas from millions of Facebook users without their consent. However, unintentional data breaches are also a major problem with series upon series of emails and passwords often being exposed from various online platforms.
- **Variability** - the inconsistency with the data. Variability is often used to refer to changing the meaning of data which is especially important when considering language processing application since the meanings of words strongly depend on the context. Variability is also found to refer to outliers and anomalies in the data. This variability can be attributed to the differences in data sources and types. Variability is sometimes even used to describe inconsistencies with speeds of Big Data streaming in databases.

When researching this topic from different sources, it can be observed that authors often substitute one or more of these terms with others of which some may not even be mentioned in this

enumeration. Also, heated discussions within the scientific community are currently in progress with the goal of determining the most crucial out of these characteristics with some even offering just *value* as the core concept that should be focused on. Nevertheless, some authors have even extended the Big Data characterization to more than ten V's (Panimalar, Shree, & Kathrine, 2017). Several of those also mentioned are: **virality** (describing the spreading speed of the data or the rate at which it is circulated between users or through a network), **viscosity** (the temporal incompatibility between sequential events or rather the time difference between an event happening and being recorded, also referred to as the resistance to the flow of data), **venue** (the variety in platforms that can be used to generate data that may result in problems with interoperability), **vocabulary** (differences in the way the data is structured), **vagueness** (usefulness or the amount of information that can be extracted from the data), **verbosity** (redundant storage of data that can lead to confusion and errors), **voluntariness** (the application of big data in an open nature to provide improvements to general human well-being, not just being business and profit oriented but also aiding healthcare institutions, public services etc.) and **versatility** (the flexibility and diversity deeply rooted within Big Data allowing it to be used in a wide range of applications).

Nevertheless, when discussing Big Data, regardless of how many V's are considered to describe it, it is clear the Big Data implementation have a wide variety of inherent obstacles that require well architected solutions in all aspects from data storage, management, processing, analysis and presentation.

2.2 Big Data Applications in General

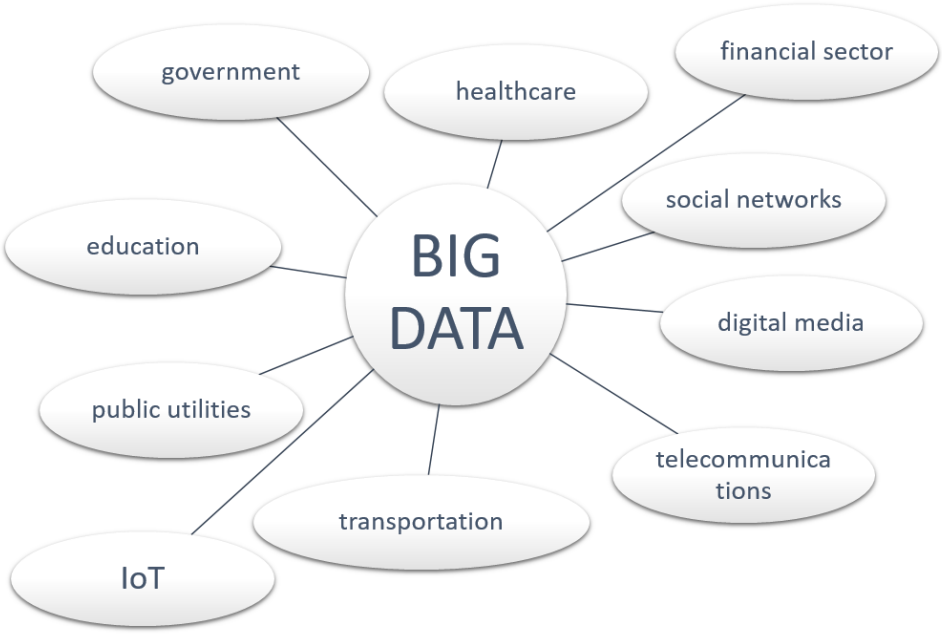

Figure 2. The general applications of Big Data

Having such a broad definition and a general idea behind it, Big Data has almost as broad of a scope of applications where it can be put to use and an even broader set of positive outcomes that can be achieved using it. This concept was, just like many other contemporary solutions used every day, originally developed as a software engineering paradigm, but has, relatively quickly,

found its way into a wide variety of applications with the possibility of basically every part of daily life being covered by one of them.

Big Data applications can be easily found in different industries, as depicted in Figure 2. First of all, the **healthcare and pharmaceutical industries** can hugely benefit by collecting and processing the immense amounts of data that is generated every hour in those branches. At the most rudimentary level, the data collected by modern healthcare applications like Apple Health, Google Health or Samsung Health (previously SHealth) just to name a few, that almost every smartphone possesses, is huge by its quantity but also insightful. The main benefit of using such applications is that, with the adequate permissions by the user, the collection of data is automated meaning there is no need for tedious data labelling which plagues many potential sources of useful information. In general, the data obtained by wireless body area networks (WBANs) can be hugely beneficial in this regard when correctly processed. Furthermore, professionally collected data based on clinical trials, patient records, various test results and other similar data can be, or rather already is in some cases, stored, processed and analysed in order to improve the effectiveness of medical institutions and aid doctors with their difficult task of decision making. Advanced data management in this field can also improve patient care with better prescription recommendations and optimizations can be applied to the drug supply chain, in turn, cutting losses and increasing efficiency. The pharmaceutical industry employing Big Data can also design targeted ad campaigns to boost sales. Discovering disease patterns and making predictions is also made easier and, not to mention, the potential of an early epidemic and pandemic detection. (Das, Rautaray, & Pandey, 2018) outlines the general potential uses of Big Data in medicine like heart attack prediction, brain disease prediction, diagnosis of chronic kidney disease, analysing specific disease data, tuberculosis prediction, early hearth stage detection, HIV/AIDS prediction and some general aspects like disease outbreak and disease outcome prediction. (Lee & Yoon, 2017) discusses some technical aspects of big data applications in medicine like missing value, the effects of high dimensionality, and bias control. (Ristevski & Chen, 2018) mention privacy and security on the topic of Big Data in healthcare, while (Shah, Li, & Shah, 2017) talks about integrating raw medical data in to the deduction process and (Tafti, Behravesh, & Assefi, 2017) offers an open source toolkit for biomedical sentence classification. Modern concepts relating to mobile health are discussed in (Istepanian & Al-Anzi, 2018) with (Bayne, 2018) exploring Big Data in neonatal health care.

The **financial and insurance sector** can also be a great host for Big Data implementations. With the current state of the worldwide economy, bank loans and credits have become mundane everyday occurrences with money lenders and investors having a constant requirement for risk analysis and management and credit default predictions when working. As this industry generally adopts new technologies early on, there is no shortage of digital data regarding both historical proceedings and new streams of information collected from ongoing transactions happening every second. Performing inference on this data is the key for detecting fraud and security vulnerabilities with many banks already using machine learning techniques for assessing whether or not a potential borrower is eligible for a loan, what his credit card limit should be etc. Credit rating is applied for all sorts of potential users, ranging from natural persons to entire countries and thus different granularity of data should be considered based on the level of classification that is to be made. Furthermore, novel approaches state that the applied machine learning can be supplemented with semantic knowledge thus improving the requested predictions and classifications and enriching them with reasoning explanations that pure machine learning based deduction lacks (Bellomarini, Gottlob, Piris, & Sallinger, 2017). The stock market is also a considerable use case for big data as the sheer volume and frequency of transactions slowly renders traditional processing solutions and computation methods obsolete. Finding patterns and surveilling this fast-paced process is key for proper optimization and scam prevention. (Gutierrez, 2017) offers a general guide for applying Big Data in finance and (Bataev, 2018) analysing relevant technologies. (Hasan, Kalıpsız, & Akyokuş, 2017), (Huang, Chen, & Hsu, 2016) offer concrete approaches like predicting market conditions by deep learning and applying market profile theory

with (Tian, Han, Wang, Lu, & Zhan, 2015) discussing latency critical applications, (Begenau, Farboodi, & Veldkamp, 2018) looking at the link between Big Data and corporate growth and (Óskarsdóttir, Bravo, Sarraute, Vanthienen, & Baesens, 2019) placing an emphasis on data collected from social networks and mobile phones.

When considering Big Data applications, one cannot overlook the massive impact that the development of **social networks** like Facebook and Twitter had on this field of research with some of those companies even driving new innovation and playing a major role in this field. Social networks provide a source of personalized Big Data suitable for data mining with several hundreds of thousands of new posts being published every minute meanwhile being great platforms for implementing big data solutions whether it be for advertising, search suggestions, post querying or connection recommendations. The social network structure has also motivated researchers to pursue alike architectures in the Big Data domain. From related literature, (Saleh, Tan, & Blake, 2013) and (Peng, Yu, & Mueller, 2018) address challenges in social networks that can be solved with Big Data, (Persico, Pescapé, Picariello, & Sperlí, 2018) gives a performance evaluation of Lambda and Kappa architectures and (Ghani, Hamid, Hashem, & Ahmed, 2018) classifying analytics solutions in the Big Data social media domain.

Another industry where Big Data can easily find its place are **world wide web, digital media and commerce**. As with all services available to end users, the wide variety of online shopping web sites also presents a continuous source of huge volumes of data that can be stored, processed, analysed and inferred on creating recommendation engines with predictive analytics. As a means to increase user engagement, multi-channel and cross-channel marketing and analysis are performed to optimize product presence in the media fed to the user. It is no accident that a certain advertisement has started to show right after the user has searched for that specific product category that the advertised product belongs to. Examining user behaviour patterns and tendencies allows for offer categorization in the best possible way so that the right offer is presented precisely when it needs to be and thus maximizing sale conversions. Data received from Big Data analysis can also be used to govern product campaigns and loyalty programs. However, content recommendations (inferred from Big Data sources) in this domain are not only related to marketing and sales, but are also used for proper display of information relating to the user. Some search engines companies have even publicly stated that their infrastructure relies on Big Data architecture which is not surprising considering the amount of data that needs to be processed. Paper (Sun, Gao, & Xi, 2014) presents an application of widely spread recommendation systems of contemporary e-commerce systems to more traditional non e-commerce systems, (Li & Huang, 2017) talks about collaborative filtering for offer recommendation, (Shi & Wang, 2018) discusses business analytics and (Wu & Lin, Unstructured big data analytics for retrieving e-commerce logistics knowledge, 2018) generate logistical strategies from two theories: resource dependency and innovation diffusion.

Following the already mentioned impact of using smart mobile phones as data sources, the **telecommunication** industry must also be considered when discussing Big Data. Mobile, television and internet service providers have customer retention as their core interest in order to maintain a sustainable business. Therefore, in order to prevent customer churn, behaviour patterns are analysed in order to provide predictions on customers looking to switch their provider and allow the company to act in time and offer various incentives or contract benefits in due time. Also, besides this business aspect, telecommunication companies using Big Data analytic solutions on data collected from mobile users can use the information generated in this way to assess problems with their network and perform optimizations thus improving the quality of their service. Since almost all modern mobile phones rely on wireless 4G (and 5G in the years to follow) networks to communicate when their users are not at home or work, all communication is passed through the data provider's services, and in processing this data still lie many useful bits of information as only time will tell what useful applications are yet to be discovered. Papers covering this aspect include (Yazti & Krishnaswamy, 2014) and (He, Yu, & Zhao, 2016) outlining mobile Big Data analytics

while (Amin, et al., 2019) talks about preventing and predicting the mentioned phenomena of customer churn and (Liu, McGree, Ge, & Xie, 2016) talks about collecting data from mobile (phone and wearable) devices.

One of the most stressful and time consuming activities in modern everyday life is probably **transportation**. The cities we live in, especially in Europe, were usually built more than a few hundreds of years ago, when their population was only a fraction of what it is today and when owning your own means of transportation was not so common. The urbanization has taken its toll with many governments not being able to cope with the exponentially increasing number of vehicles on their roads, and large infrastructure projects generally lagging behind the increasing traffic volumes. This problem is not only present on our roads but also in flying corridors all across the world. With shipping from distant locations getting cheaper by the day, it is no wonder that the air traffic volume is also increasing at a very high rate. In the status quo, a question is posed if the traffic flow can somehow be optimized to shorten travel times, cut costs and increase logistical performances of modern day delivery systems. One of the readily available solutions to this problem is the navigation/routing program that modern day mapping applications provide. Using the Global Positioning System (GPS) data, usually from a large number of smartphone users, the routing system can decide in real time not only the shortest but also the fastest route between two desired points. Some applications like Waze rely on direct user inputs in order to locate closed off streets, speed traps etc. but at its most rudimentary level, this approach can work with just raw GPS data, calculating average travel times per street segments, and thus forming a live congestion map. Of course, such a system would be of no benefit to end users if it were not precise, but since the aggregated results that are finally presented are obtained based on many different sources, classifying this as a Big Data problem, the data uncertainty is averaged out, and an accurate result can be output. Self-driving cars also rely on vast amounts of data that is constantly being provided by its users and used for training the algorithms governing the vehicle in auto-pilot mode. Holding on to the automation aspect, Big Data processing in the transportation domain could even be used to govern traffic light scheduling which would have a significant impact on this sector, at least until all vehicles become autonomous and traffic lights are no longer required. Furthermore, the transportation domain can be optimized using adequate planning obtained from models with data originating from user behaviour analysis. Ticketing systems in countries with high population density or frequent travellers where reservations have to be made, sometimes, a few months in advance, rely on machine learning algorithms for predictions governing prices and availability. Patterns discovered from toll collecting stations and border crossings can be of huge importance when planning trip durations and optimizing the selected route. The collection of related articles to this topic is possibly the largest of all applications. (Zhang, Jia, & Ma, 2018) offers a methodology for fare reduction in modern traffic congested cities, (Liu D. , 2018) discuss Internet-of-Vehicles, (Grant-Muller, Hodgson, & Malleson, 2017) talks about the impacts that the data extracted from the transport domain has on other spheres, (Wu & Chen, Big data analytics for transport systems to achieve environmental sustainability, 2017) mention environmental effects, (Torre-Bastida, Del Ser, & Laña, 2018) talk about recent advances and challenges of modern Big Data applications in the transportation domain while (Imawan & Kwon, 2015) analyses the important concept of visualization in road traffic applications. Also related, (Ghofrani, He, Goverde, & Liu, 2018) surveys Big Data applications for railways, (Zeyu, Shuiping, Mingduan, Yongqiang, & Yi, 2017) and (Gohar, Muzammal, & Ur Rahman, 2018) discuss data driven modelling in intelligent transportation systems and (Wang, Li, Zhou, Wang, & Nedjah, 2016) attempts fuzzy control applications in this domain.

An unavoidable topic when discussing Big Data applications, in general, is the **Internet of Things** (IoT) and home automation. As modern sensing equipment is capable of running at second and sub-second sample periods, the sheer volume of data being generated by smart home devices today is already inconceivable, and will only exponentially grow as more residents adopt these new technologies and they become more commonplace. Home automation solutions based on this data must be capable of proper processing and providing accurate predictions as not to interfere with

our daily routines but rather be helpful and truly save time. These solutions can also offer energy saving recommendations, present optimal device scheduling to maximize comfort and minimize costs, and can even be extended from the operation aspect to planning and offer possible home adjustments or suggest investments in renewable sources if the location being considered is deemed fit. Having smart appliances initially presented the concept of human-to-machine communication but, governed by Big Data processing, this concept is further popularized with machine-to-machine communication where the human input is removed resulting in less interference. Predictive maintenance and automatic fault detection can also be obtained from sensor data for both basic household appliances and larger mechanical systems like cars, motors, generators etc. IoT applications require proper cloud frameworks as discussed in (Wang & Ranjan, 2015), with related challenges depicted in (Kaul & Goudar, 2016). (Ge, Bangui, & Buhnova, 2018) presents a comprehensive survey of Big Data applications in the IoT sphere, (Ahmed, et al., 2017) talks about analytics, (Martis, Gurupur, Lin, Islam, & Fernandes, 2018) introduce machine learning to the mix. (Kumari, et al., 2018) also gives a survey but with a main focus on multimedia and (Kobusińska, Leung, Hsu, Raghavendra, & Chang, 2018) talks about current trends and issues.

Public utilities can also rely on Big Data to improve their services. Advanced water supply system with proper flow sensing equipment can, in real time, detect leakages or illegal connections. These types of problems also plague communities with district heating systems where hot water is supplied to end users for heating through a pipeline. Especially in winter, when hazardous situations on these supply systems like leakages or pipe malfunctions are common, whole neighbourhoods to be cut off from the heating supply for several days and weeks until the problem is fixed. Therefore, having good predictive capabilities and quick fault detection is key for maintaining a stable system. Smart grids are also often mentioned as good cases for Big Data implementations as the national electric power supply depends on the good grid infrastructure. Having proper metering equipment and sensors can allow for quick and easy interventions if need be but also provide with key insight on load distribution and profiles required by plant operators for sustaining system stability. Predictive maintenance also plays a key role in smart grid upkeep since all of its segments are both critical and expensive, and any unplanned action cuts users from the electricity supply upon which almost all modern device rely to function. (Zohrevand, Glasser, & Shahir, 2016) talks about the application of Hidden Markov models for problem detection in systems for water supply, (Bharat, Shubham, & Jagdish, 2017) also discussing smart water systems while (Ku, Park, & Choi, 2017) analyses microgrid IoT applications. Related IoT applications regarding water supply are found in (Koo, Piratla, & Matthews, 2015), while, on the other hand (Tu, He, Shuai, & Jiang, 2017) reviews issues with Big Data implementations in smart grids and finally (Munshi & Mohamed, 2017) and (Phan & Chen, 2017) talking about analytics and monitoring for smart grids.

Another important branch where Big Data can be made use of is **education.** Performing analytics on past records of student performance where huge information volumes are available to allow precise predictions of future results, easier selection of favourite subjects and curriculum activities. With each pupil leaving a significant trail of relevant information during his education process, this field of application offers vast potential. In close relation are career prediction solutions that can aid and guide students when making very important life-changing decisions regarding university selection, study programs etc. Customized learning is especially discussed in online and distance learning applications because of the inherently digital nature of related data. Also, Big Data processing is used to reduce dropouts. The appeal of Big Data in education is shown in research papers on this topic where (Matsebula & Mnkandla, 2017) discusses analytics for higher education, (Li, Li, & Zhu, 2017) quality monitoring and evaluation, (Nelson & Pouchard, 2017) a modular curriculum, (Liang, Yang, & Wu, 2016) dropout prediction, (Yu, Yang, & Feng, 2017) online education and (Qiu, Huang, & Patel, 2015) specially focuses on assessing higher education in the USA. On the other hand, (Schwerdtle & Bonnamy, 2017) focuses on nurse education, (Pardos, 2017) talks about sources of Big Data in education and models for standardized tests, (Li & Zhai, 2018) offers a review of modern Big Data in education, (Santoso & Yulia, 2017) focus specifically

on data warehouses and (Olayinka, Kekeh, Sheth-Chandra, & Akpinar-Elci, 2017) goes back to health education applications

Finally, Big Data can also be employed in both **government** and **political** services. Data mining in this regard are often used to infer political trends and general attitude of the population. Some countries even practice deriving legislation from principles recommended and voted by individual citizens. On the other hand, analysis can also be performed in order to predict potential crime sites with risk-terrain modelling and pre-emptively inform police on problematic regions. Similar techniques can also be applied to traffic, determining dangerous road sections causing frequent accidents. Interoperability between different government agencies is also mentioned as an example of a system that can be accelerated by using Big Data as readily available data can save substantial amounts of time when data querying. Concrete work on this topic is done by (Aron, Niemann, & ., 2014) giving best practices in governments, (Yan, 2018) gives a general perspective of Big Data applications in this field, (Sixin, Yayuan, & Jiang, 2017) focuses on ecological environment in China, (Hardy & Maurushat, 2017) talks about driving innovation by opening data up in the public domain and (Jun & Chung, 2016) examines electronic governance in Korea.

3. Landscape

3.1 Big Data Frameworks

Ever since its introduction back in 2008, the **Hadoop** system currently supported by Apache has basically become the standard framework for Big Data implementations. This can mainly be attributed to its reliability, robustness, high scalability potential and parallelism. Also, a thing to note is that **Hadoop** can run on distributed commodity hardware of different sources, and so significantly decreasing costs of eventual system upgrades and maintenance. This framework was much quicker than traditional solutions because it was able to quickly process extremely large datasets since operations could be executed locally, on machines where the data is stored, contrary to the standard solutions that had to copy the entire datasets into the main processor memory. **Hadoop** introduced fault-tolerance to overcome failures which are a common occurrence in distributed systems through replicating data amongst different nodes further supplementing its quick nature. **Hadoop** relies on its file system, **HDFS,** and the **MapReduce** processing method in order to work properly. **HDFS,** as a high-latency master-slave batch processing system, was designed for huge volumes of data and thus supports clusters with hundreds of nodes with resilent interoperability between distributed heterogeneous hardware. The **MapReduce** programming model, on the other hand, presents the idea of splitting input data into distinct patches, processing them (assigning values in key-value pairs), reordering data and finally merging it in a two-step process with steps called mapping and reducing, resulting in the given name. Solutions like **YARN** for cluster managing and **HBase** (which will also be discussed later in this report) for database management are often associated with **Hadoop.** These solutions often include their own libraries for performing inference on the data (like machine learning) and **HBase** is no exception with support for **Mahout.** Frameworks for Big Data applications also worth mentioning are **Apache Spark, Storm, Samza** and **Flink,** summarized in Table 1. All of the mentioned frameworks, like **Hadoop**, are open source, they generally support Java and are **HDFS**.

Streaming solutions mentioned in Table 1 offer lower-latency processing for quicker results with **Spark** being a hundred times faster than **Hadoop** in some applications but in turn consuming more memory. The intrinsic concept in **Spark** making this possible is its resilient (fault-tolerant) distributed (data spread across nodes in a cluster) dataset (partitioned data) (RDD) supporting two basic operations called transformations (creation of new RDDs from existing ones) and actions (final RDDs computation). Some **Spark** APIs worth mentioning are **SparkCore, SparkStreaming, SparkSQL, SparkMLLib** and **GraphX. Spark** and **Flink** share similarities in efficient processing in graph-like structured Big Data and are both enriched with APIs for machine learning, predictive and graph stream analysis. As working with heterogeneous data is a top priority, **Hadoop, Storm** and **Flink** all use key-value pairs when representing data and all five frameworks assume a cluster architecture whilst having specific cluster managers per architecture.

Storm is a fast, streaming, real time suitable framework designed to work with both structured and unstructured data. The arrangement of a **Storm** structure can be described as a directed acyclic graph with nodes defined as either spouts (data sources) or bolts (operations that should be performed). Since bolts are distributed across multiple nodes in **Storm,** parallel processing can be achieved. **Samza** was also conceived as a fast streaming real-time ready processing platform that was intended for use in data ingestion, tracking and service logging with the ability to operate on large messages and offer persistence capabilities for them. **Kafka** and **YARN** are used within **Samza** to facilitate its operation. **Samza** is often referred to as a three layered structure with the first layer designated for data streaming employing **Kafka**, the second managing resources with **YARN** and the third in charge of processing. Finally, **Flink** allows for both batch and stream processing with a similar programming model to **MapReduce** but with **Flink** offering advanced operations like join, aggregate and filter. Its machine learning library, **FlinkML** gives an ability of efficient and scalable processing of high data volumes. Similar to aforementioned frameworks,

Flink uses a DAG structure representing submitted jobs and after processing those jobs, requests are passed on to lower levels housing cluster managers and storage controllers.

On top of mentioned frameworks, novel solutions are being developed, such as **SANSA** (Semantic Analytic Stack) (Lehmann, et al., 2017) based on Spark and Flink, which is a Big Data engine that incorporates the processing and analytics tools best suited for RDF data. It offers easy integration with other solutions and a variety of features through the inclusion of a native RDF read/write, query, inference and machine learning libraries.

Table 1. Big Data frameworks and their characteristics (Inoubli, Aridhi, Mezni, Maddouri, & Nguifo, 2018), (Oussous, Benjelloun, Lahcen, & Belfkih, 2018)

	Hadoop	Spark	Storm	Samza	Flink
Foundation	Apache Software Foundation	UC Berkeley	BackType, Twitter	LinkedIn	Apache Software Foundation
Supported language	Java	Java, Python, R, Scala	Any	Java	Java
Data sources	HDFS	HDFS, DBMS and Kafka	HDFS, HBase and Kafka	Kafka, Kinesis	Kafka
Execution model	batch	batch, streaming	streaming	streaming	batch, streaming
Programming model	Map and Reduce	Transformation and Action	Topology	Transformation	Map and Reduce
Cluster menager	YARN	Standalone, YARN and Mesos	YARN or Zookeeper	Zookeeper	YARN
Machine learning compatibility	Mahout	SparkMLib	SAMOA API	FlinkML	SAMOA API
Fault tolerance	Duplication feature	Recovery technique on RDD objects	Checkpoints	Checkpoints	Data partitioning
Operating system	Cross-platforms	Windows, macOS, Linux	Cross-platforms	Cross-platforms	Cross-platforms

3.2 NoSQL Platforms and Knowledge Graphs

As was previously mentioned, Big Data usually involves simultaneous processing of heterogeneous data from a variety of different sources, and thus that data generally differs both

from source to source and in time. Having a structured, semi-structured and not structured at all datasets as a deluge of data formats that needs to be worked on in high volume and rate requires the use of advanced, flexible and scalable solutions.

To adequately answer the requests for data organization posed by the Big Data paradigm, traditional relational database management systems (RDBMS) are usually replaced by modern NoSQL solutions. Although sometimes misused as an acronym for systems having (absolutely) no SQL in processing, NoSQL is actually an abbreviation of “not only SQL” meaning that NoSQL systems actually sometimes employ SQL but supplement it by other technologies. These technologies, outlined in (Patel, 2016) and (Leavitt, 2010) can be more accurately referred to as non-relational databases (as opposed to RBDMS) because they omit the use of tables as a means for data storage in the traditional way and such solutions offer flexible environments that can be used for Big Data and real-time applications. Note that a recent trend in systems traditionally known as NoSQL systems has started that adds SQL support for such systems.

Being able to work with a variety of data models in NoSQL means that there is no predefined schema for structuring data, no tables and columns need to be defined, and in some cases, NoSQL can even work with completely dynamic data with its structure constantly changing. One of the most important advantages of NoSQL over RDBMS is the ability of horizontal (outward) scaling as opposed to vertical (inward) with the differences illustrated in Figure 1. RBDMS requires many more local resources leading to larger higher limitation and costs, while NoSQL is more optimized for distributed computing on distant nodes or computer clusters. The join operation, essential to RBDMS is very slow and difficult to perform across distributed networks. This difference in scaling is especially important in discussed applications because the volume of data is constantly increasing. Furthermore, RBDMS can result in highly complex, large and slow to process structures when the data that needs to be stored is not natively in the appropriate format because of the conversions that need to be applied. SQL itself can also sometimes be considered an issue because, although its structure can be very efficient in highly structured data applications, it is considered far too rigid for the flexibility required for Big Data. NoSQL also allows for flexible fault tolerance policies that sometimes put speed ahead of consistency while the traditional RBDMS are inherently ACID complaint and all queries and requests must be processed in their entirety and without errors before a result is displayed to the user.

NoSQL database solutions can be categorized according to their data storage types, as is given in (Corbellini, Mateos, Zunino, Godoy, & Schiaffino, 2017). The first is the key-value category where related literature places **Hazelcast, Redis, Membase/Couchbase, Riak, Voldemort** and **Infinispan.** Another approach is the wide-column format where **Apache HBase, Hypertable** and **Apache Cassandra** are classified. Data can also be saved in a document-oriented manner where **MongoDB, Apache CouchDB, Terrastore** and **RavenDB** cover most of the market share. Finally, NoSQL databases can be graph-oriented like in, for example, **Neo4J, InfiniteGraph, InfoGrid,**

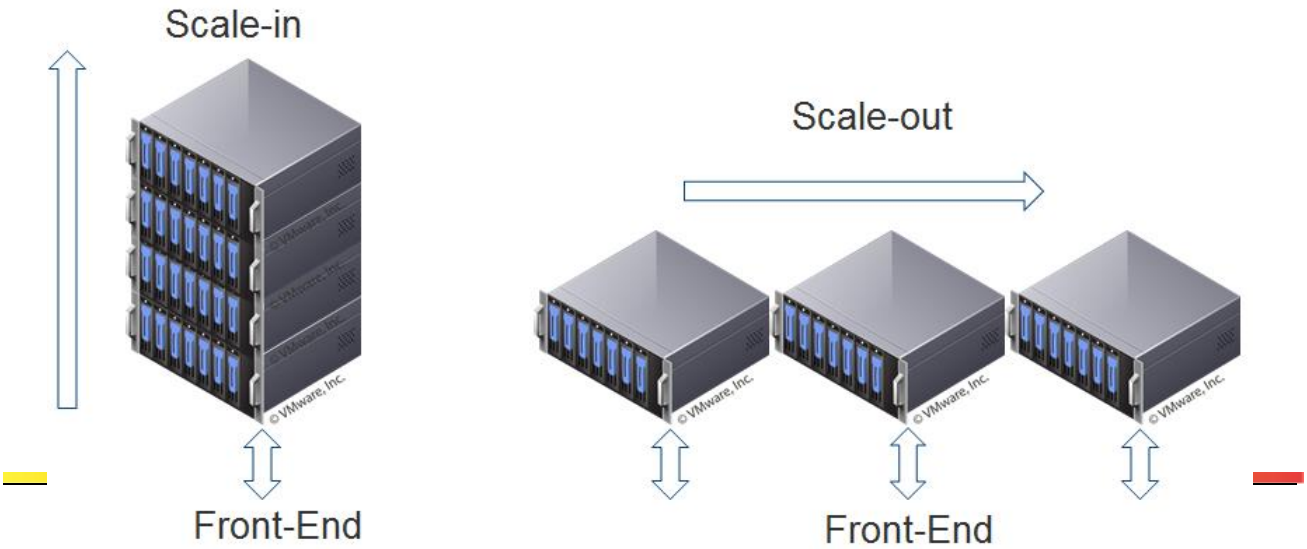

Figure 3. Two basic types of scaling, source

HypergraphDB and **AllegroGrap** and creatively named **BigData.** Some of these solutions are reviewed in depth with hands-on experience and screenshots in (Rai & Chettri, 2018).

Key-value stores are simple database concepts that record data as key-value pairs that support at least two basic operations being key retrieval (get) from and key-value saving (put) to the database. They offer consistent hashing and vBuckets as a means for sharding, virtual bucketing, and other dynamo-based mechanisms (vector clocks, sloppy quorum and hinted handoff and Markle trees) stemming on the **Amazon Dynamo** used for the shopping cart and session management services at Amazon.

Wide column databases (also known as Column family database) introduce the concept of storing data in columns where some rows can but need not have all of the columns of a certain type, lessening the rigid nature of traditional table based approaches. Also, this structure allows for data compression by appropriate algorithms because the data stored in a selected column is of the same type. The original motivation behind this approach was the **Google BigTable** database on top of Google's File System (GFS) created for storing petabytes of data originating from Google Earth, Google Maps, Blogger and some other services. This solution also supports versioning with its addition of the timestamp dimension. Some column family databases like **HBase** run on top of previously mentioned technologies like HDFS and employ MapReduce algorithms to parallelize workloads while **Cassandra** uses the aforementioned dynamo-based features to manage storage and replication. All cited solutions support querying trough Pig and Hive while **Cassandra** and **Hypertable** offer their CQL and HQL query languages respectively. However, it is worth mentioning that column family database solutions are relatively rare since their applications tend to be very specific and do require some structure to the data or high complexity is risked, similar to RBDMS.

On the other hand, document-based (document-oriented) stores save data as collections of documents as opposed to structured tables with uniform fields per record. This structure allows for adding an arbitrary number of fields with arbitrary lengths whenever necessary. These databases can also be abstracted as somewhat more complex key-value stores where some structure is attributed to the value from the data pair and are used when the number of fiends in that structure cannot be exactly determined during the software development process. The data generally encoded in databases from this category is semi-structured and an industry standard format like XML, JSON and BSON is used to represent the data. This structure allows for the addition of features that were previously unavailable in key-value databases. From the list of most frequently used document-based databases, **MongoDB** is a cross-platform free solution using binary encoded JSON objects (BSON), **GridFS** for large file storage and **Mongo** for routing queries, and supports consistent atomic operations on document level. **CouchDB** offers ACID compliant transactions per document in a consistent way and employs multiple version concurrency control. Some of the most popular **CouchDB** distributions are **BigCouch, Lounge** and **Pillow. Terrastore,** based on **Terracotta**, follows the JSON standard and supports either client connections from Java, Scala and Clojure or HTTP access. Again, consistency is implemented on a document level. Lastly, **RavenDB** also supports ACID transactions but does not use locks and Querying is done through a special language LINQ alike SQL. **RavenDB** stands out with its customizable document sharding. Using markup languages like XML for encoding data allows for the use of domain-specific querying language for such implementations. In this regard, **XPath** is used for querying nodes with **XQuery** (XML Query Language) building on top of it and allowing for querying XML data in a manner similar to SQL as well as data manipulation. The use of both of these standards is supported by the W3C. Furthermore, **XSPARQL** was introduced to bridge the gap between raw XML data as an often used storage format and RDF formatted data which is especially important in the graph applications regarding semantic data in general and the semantic web applications concretely. Lastly, **OXPath** was created with the ability of interactions with sophisticated interfaces in web applications, precisely gathering data, good scaling capabilities and an embedded API for

seamless integration with other existing technologies. It is extended from **XPath**, able to handle XML, JSON, CSV and parsing this data into RDBMS.

Finally, graph-oriented databases begun to be used by social network and map developers without a reference design solution, leading to a wide variety of available solutions today. Graph databases represent data as containers with the originating vertex of the graph and pointers to the connected vertices with a distinction made between commutative (undirected) and non-commutative (directed) edges of such structures. Using MapReduce in a graph-oriented database has proved problematic because the structure of the graph is not known a-priori and so processing frameworks like **Pregel, Apache Giraph, Trinity, HipG** and **Mizan** were developed to cope with this issue. Besides the solutions mentioned in this category, some application-specific solutions worth mentioning are Twitter's **FlockDB** and **Graphd.**

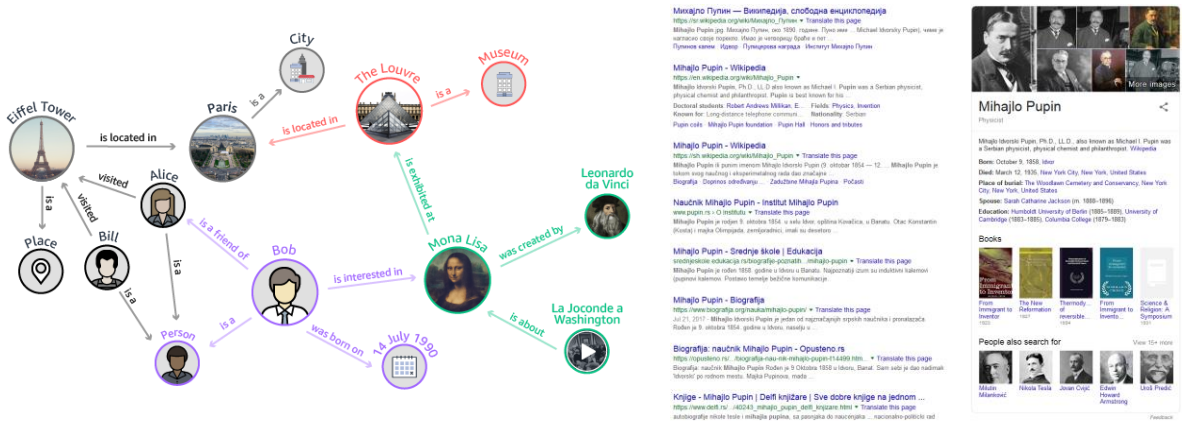

Figure 4. An example of a knowledge graph and how data derived from it can be represented, source1

In modern data driven enterprises, the idea of just having a collection of data that is stored in a (graph) database is simply not enough because this data is generally unconnected and in itself does not carry any knowledge. Storing data over time without performing any useful manipulation over it leads to having large data lakes and warehouses of unused, unorganized data, often from multiple sources. However in particular in scenarios that consider "world" Knowledge Graphs, huge benefits can be drawn from representing data in structured graph form presented by data triples: subject, predicate and object (entity, property and value, RDF) where subjects and objects are placed in the nodes of the graph and their relationship is denoted on the connecting edge, because information in these types of environments is generally most suitable for relational graph-like representation. A simple structure of this form is shown in Figure 4. Also, not being tied to the fixed structure given by RDBMS allows for flexible schemas where an arbitrary number of nodes and relations can be added at will without compromising the underlying architecture. Where traditional solutions would require having simultaneous copies of data, knowledge graphs do not have this

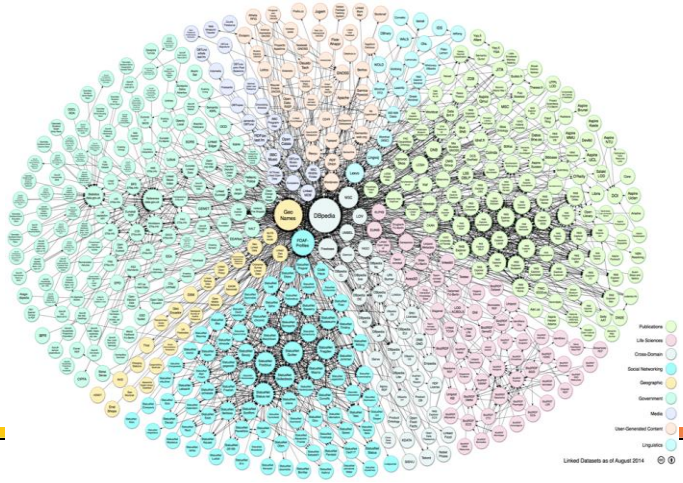

Figure 5. The knowledge base behind Dbpedia, source

ambiguity problem. Therefore, knowledge graphs are already used by Google, Facebook, Microsoft (and LinkedIn), Amazon and are slowly making its way into finances, sales and other aspects of daily life. Many companies have their own proprietary knowledge graphs nowadays. For example, Google now supplements their sorted search results with a table like structure to the right side where related structured data is extracted from their knowledge graph and displayed to the user, as can be seen in Figure 4. Similar recommendation engines are also used when presenting search and friend suggestions, related videos, finding links between entities etc. There are, of course, open repositories of structured data for building knowledge graphs with **DBpedia, Wikidata** and **YAGO** leading in this category. The knowledge resource base behind **DBpedia** is presented in Figure 5. Note however, that graph databases (such as Neo4j), world Knowledge Graphs (such as DBpedia, Wikidata and Yago) and Enterprise Knowledge Graph Management Systems are often confused, but typically refer to very different kinds of systems in what features they offer and how they are intended to be used.

3.3 Stream Processing Data Engines

In order to be able to cope with complex and dynamical information, the first widely spread and used advancement based on SPARQL was proposed as **Continuous SPARQL** (**C-SPARQL**) in (Barbieri, Braga, Ceri, Valle, & Grossniklaus, 2010). Namely, its main contributions were in stream organization, providing continuous queries and supporting aggregation. With streams representing sequences of RDF triples, RDF stream was defined as an ordered pair of an RDF triple and its timestamp, which were monotonically non-decreasing due to the time flow. Moreover, by including RDF's expiring date paradigm, it quickly and efficiently removes the invalid data. Nevertheless, **C-SPARQL** is not designated for extensive amount of static data, and this is one of its primary drawbacks. Another SPARQL extension proposed in literature is **Streaming SPARQL** which enabled managing RDF based data streams by transforming queries into the new proposed expanded algebra (Bolles, Grawunder, & Jacobi, 2008).

Furthermore, a SPARQL addition, written in Prolog, with the main contribution in event processing and stream reasoning, is proposed as **Event Processing SPARQL** (**EP-SPARQL**) in (Anicic, Fodor, Rudolph, & Stojanovic, 2011), and in combination with the ETALIS Language for Events, **ETALIS** engine is developed and presented in the same paper. It was used for real-time complicated discovery of troublesome events. Contrary to others, it facilitates both deduction on temporal and static knowledge and efficient event processing. Nonetheless, it is worth highlighting, that all of previous approaches expand upon SPARQL by using sliding windows for RDF stream processing. More on their comparison, as presented in (Kharlamov, 2017), contrary to other, as **EP-SPARQL** is primarily focused on events, it does not facilitate temporal windows. Finally, it can be concluded that all three of them are based on RDF streams, executed in a different manner, from using physical streaming algebra, DSMS and logic programming, with no cascading paradigm supporting in any of them. In (Le-Phuoc, et al., 2012), the Java wrapper for the **ETALIS** named **JTALIS** is proposed.

Another innovative and extensively used solution for managing streaming data over **Linked Stream Data is Continuous Query Evaluation over Linked Streams** (**CQELS**) (Le Phuoc, Dao-Tran, Xavier Parreira, & Hauswirth, 2011), which primarily contributes with its "white box" approach. Contrary to **C-SPARQL**, **SPARQL** and **EP-SPARQL** which pass on the query processing to the alternative engines, **CQELS** performs it natively, as its main improvement to stream data processing engines. It is stated that the content-change policy enabled, implying that queries are generated at the very same moment the new statements appears (Ren, Khrouf, Kazi-Aoul, Chabchoub, & Cure, 2016). According to the same paper, where **C-SPARQL** and **CQELS** were compared as native, **CQELS** offers fewer operations. Moreover, when the multi-streams are to be concerned, the **CQELS** turned up to be less reliable, due to its asynchronous streams. Nonetheless, **CQELS** is stated as preferable when simple queries and static data is considered, because of **C-SPARQL**'s scalability problem with static data.

As a **CQELS** enhancement, the **CQELS+** is presented in (Van, Gao, & Ali, 2017), with its contribution being the shared join based operator. Moreover, further performance encouragement is achieved by enabling multiple query processing and load balancing so as to optimize parallel efficiency.

In order to create an incremental stream data processing engine, **INSTANS** was proposed in (Rinne, Nuutila, & Torma, 2012). It is based on Rete-algorithm, stated as near-real-time with the ability of concurrently processing multiple queries, which distinguishes it from all of the others.

Finally, to facilitate the capabilities of performance measuring and comparing previously proposed stream engines under some common criteria, a set of benchmarks is proposed in the literature as shown in

Table 2. **SRBench** was the first presented one, incorporating real data from the Linked Open Data cloud, intending for collecting performances on descriptive and realistic real world examples. It was designed for performance testing on highly convoluted, changeable, realistic data in combination with static, so as to adequately depict engine's performance both on simple and highly complicated queries. Furthermore, another proposed approach was **LSBench**, with its main goal to run and compare proposed engines on real world data, which was realized by the presented social network data simulator, as one of the main application fields of these engines. Similarly to the **SRBench**, the first test was tended for functionality validation, whilst the remaining two ones were intended so as to verify the output's faultlessness and the maximum amount of input RDF streams that can be processed by the engines, as stated in (Dell'Aglio, Balduini, & Della Valle). In order to supplement previously proposed benchmarks, as the **SRBench**'s development with an analysis of operational semantics the **CSRBench** was proposed. It analyzes the query's faultlessness, as well as the outputs validity. However, it can be stated that its most significant drawback is the correctness inclusive interpretation over time.

Moreover, as common in the smart city domain of Big Data applications, data characteristic and the application specification might change during time, which is not taken into consideration by previously presented benchmarks. Therefore, **CityBench** was presented, with its main contribution to performance estimation of engines in the smart city domain with reliable and realistic real world data. Nevertheless, its major handicap is considered to be non-considering of some substantial aspects as the stream rate. As a combination of **LSBench** and **CityBench** benchmarks, **RSPLab** was proposed, in order to cover different performance characteristics, so as to bring as much comprehensive result as possible. Finally, the newest one proposed in the literature is **YABench** (**Yet Another RDF Stream Processing Benchmark**), was intended to contribute by improving all of the mentioned segments. It enabled execution method differentiation, throughput and correctness over time, as its main contribution.

Table 2. Big Data benchmarks, (Lachhab, Bakhouya, Ouladsine, & Essaadi, 2016)

Benchmark	Paper	Year	Contributions
SRBench	(Zhang, Duc, Corchno, & Calbimonte, 2012)	2012	the first proposed benchmark with representatively and wide-purposely selected 17 queries processed on compound of dynamically changing and static data, both on single and multiple queries; no time, memory and scalability tests included
LSBench	(Le-Phuoc, et al., 2012)	2012	running engines, real data, output comparison, social network data generator; functionality, output accuracy and the maximum input throughput; variety of both data and queries included
CSRBench	(Dell'Agilo, Calbimonte, Balduini, Corcho, & Della Valle, 2013)	2013	SRBanch derivation; query and output's correctness evaluation;
CityBench	(Ali, Gao, & Mileo, 2015)	2015	realistic an reliable practical data form the smart city domain;
RSPLab	(Tomasini, Della Valle, Mauri, & Barambilla, 2017)	2017	a combination of LSBench and CityBench; intended to gather as many different estimated performances as possible
YABench	(Bozzon, Cudre-Maroux, & Pautasso, 2016)	2016	enables differing of execution method; the output performance metric is given as precision and recall for windows; enables throughout and scalability analysis, as well as the engine's correctness

3.4 Big Data Preprocessing

As consequence of many different challenges, due to enormous rapidly increasing amount of data, its uncertainty and heterogeneity, the data preprocessing in unavoidable step in the Big Data frameworks. Hence, many diverse techniques are included in this process (Garcia, Ramirez-Gallego, Luengo, Benitez, & Herrera, 2016):

- **data cleaning** - the processes of reduction of defective data, so as to improve fault tolerance. Most of the used approaches are based on machine learning so as to retrieve informative knowledge, whilst discard the unnecessary one. One of the most frequently used is dimension reduction (Zhang, Xiong, Gao, & Wang, 2017) (Tsuge, Shishibori, Kuroiwa, & Kita, 2001)
- **data normalization** - as the consequence of data heterogeneity and the prevention from the divergence of some learning algorithms
- **data transformation** - necessary in some cases for improving correctness, it combines already cleaned data in order to increase benefit of the acquired data.
- **missing values imputation** - inevitable step in all application with real data, in order to appropriately use any analytical algorithm. This step is obligatory in order to have reliable

conclusions. Therefore, many different approaches have been proposed in literature. The simplest one is discarding the missing data. Nonetheless, it is usually unsatisfactory, leading to the necessity of some more sophisticated approach. The clustering can be used in order to solve this problem (Hathaway & Bezdek, 2002) (Leng, 2014)

- **data integration and data exchange** - the grouping of the similar data, as a help for more accurate knowledge pulling out.
- **noise identification** - unavoidable in order not to inserting additional uncertainty and mistakes

In order to deal with discussed difficulties, the preprocessing frameworks **NADEEF**, **Dedoop**, **BigDansing**, **Pixata**, **Trifacta**, **Dataiku** were proposed (Tang, 2014) (Dellachiesa, et al., 2013) (Kolb, Thor, & Rahm, 2012) (Khayyat, et al., 2015).

3.5 Big Data Analytics

As previously deliberated, Big Data is intended for the processing and analysis of the giant amount of data. Hence, it is indisputable that algorithms proposed for the data reasoning can be stated as the core of this paradigm. Namely, it is imminent that all of the other mentioned components are unavoidable, yet proposed so as to serve the leading component - the knowledge tracker. Accordingly, this section is written in order to elaborate the current state of the art on analysis algorithms and their corresponding tools.

All of these algorithms can be covered by one name as machine learning algorithms. Nonetheless, the diversity amongst them is extensive, and therefore, they are going to be analyzed within the coherent appropriate groups. Firstly, depending on the main learning algorithm, they can be categorized as **supervised**, **unsupervised** and **reinforcement learning**.

Contrary to unsupervised, the supervised learning algorithms are making use of labeled information on the desired output, with the goal of minimizing error on the training data set. As classified in (Dey, 2016) and (Sharma & Kumar, 2017) most frequently used supervised techniques are **linear regression** as an approach for estimating continuous output with the presupposed model linear with the respect to the predictors, **logistic regression** as the classification approach that estimates the probability of the input affiliation to the considered class, **support vector machines** and its improvement for regression problems **support vector regression** based on the margin calculation, **Gaussian process regression** as a non-parametric statistical approach used for the regression problems, **discriminant analysis** and **naive Bayes** as, also statistical methods, yet used for classification, **neural networks**, which are, nowadays, widely spread due to their ability to extract highly informative features even from extraordinary complex problems, **ensemble methods** are proposed so as to improve performances by using combination of various of models, **decision trees** as the approach preferable for classification purposes designed in such a way that the nodes represent the attributes, whilst the branches depict their values etc. On the other hand, commonly used unsupervised approaches are **K means**, **K Medoids**, **fuzzy C Means**, **hierarchical** and **Gaussian mixture**. As the clustering methods are concerned, **Hidden Markov Models** and their advancements are broadly used as probabilistic techniques based on the Markov chain, **unsupervised neural networks** from the similar reasons as already mentioned etc.

In order to present adequately the analytics in Big Data, not only should the algorithms be presented, but also the tools containing or using them. Accordingly to a recently conducted survey in (Saggi & Jain, 2018), the mentioned approaches are discussed by the framework point of view.

As was expected, the basic concepts of classification and regression are covered by most of the frameworks - **Apache Spark**, **Rhadoop**, **Apache Mahout**, **H2O**, **R**, **MOA**, **VowpalWabbit**, **TensorFlow**, **BigML**, **Weka**. In **Apache Spark** logistic regression, decision tree classifier, random

forest, gradient-boosted tree, multilayer perceptrons, linear SVM, one-vs-all and naive Bayes are the supported concepts, whilst it implements generalized linear regression, decision tree, random forest and gradient-boosted tree for regression purposes, survival and isotonic regression as well. **H20** covers deep learning neural networks, distributed random forest, isolation forest, generalized linear model, gradient boosting machine, naive Bayes, stacked ensembles and XGBoost, whilst **R** (Prakash, Padmapriy, & Kumar, 2018) supports logistic regression, linear, mixture, quadratic and flexible discriminant analysis, neural networks, SVM, naive Bayes, kNN, and a couple of tree oriented classification methods - regression trees, random forest, gradient boosted machines, bagging CART and C4.5. Moreover, in **MOA**, available approaches for classification and regression are Bayesian classifiers, decision trees, perceptrons and drift classifiers which are proposed for data's changing nature over time, while **BigML** offers decision trees, random forest, boosting ensembles, logistic regression, deep nets. **Weka** supports linear and logistic regression, naive Bayes, decision trees, kNN, SVM, neural networks, random forest, C4.5. The proposed comparison is presented in Table 3, giving a systematic review on the topic. Also worth mentioning is Spark's **MLLib** library while Python's **Scikit-Learn** is also widely used when machine learning problems are considered.

Table 3. Systematization of regression and classificaiton learning algorithms in Big Data tools

	Apache Spark	H2O	R	MOA	Scikit - Learn	Bigml	Weka
linear regression	+				+	+	+
logistic regression	+		+		+		+
SVM	+		+		+		+
naive Bayes	+	+	+	+	+		+
discriminant analysis			+		+	+	
survival regression	+						
isotonic regression	+				+		
decision trees	+		+	+	+	+	
random forest	+	+	+		+	+	+
gradient boosting tree	+	+	+		+	+	
isolation forest		+			+	+	
bagging CART			+				
C4.5			+				+
generalized linear model		+			+		

ensembles		+			+		
XGboost		+					
NN	+	+	+	+	+	+	
kNN			+		+		
drift classifier				+			
model-fitting					+		

In addition, when clusterization is concerned, **Apache Spark**, **Rhadoop**, **Apache Mahout**, **H2O**, **R**, **MOA**, **TensorFlow**, **BigML** and **Giraph** are among suitable choices. Namely, when **Apache Spark** is considered in context of clusterizing, it implements K-means in the original form, and its bisecting and streaming improvements, Gaussian mixture, hierarchical optimization algorithms, power iteration clustering and latent Dirichlet allocation approaches are available, **R** includes K-means, Partitioning Around Medoids, clustering large applications, Fuzzy clustering, Model-based clustering and hybrid approaches, whilst **BigML** offers K-means and G-means. **Giraph** contains implemented affinity propagation and K-mean algorithms. Presented clustering tools are shown in Table 4.

Table 4. Systematization of clusterization learning algorithms in Big Data tools

	Apache Spark	H2O	R	Giraph	BigML
K-means	+	+	+	+	+
G-means					+
Gaussian mixture	+				
PIC	+				
LDA	+				+
aggregator		+			
PAM			+		
CLARA			+		
Fuzzy clustering			+		
Model-based			+		
Hierarhical	+		+		
Dencity based			+		
afinity propagation				+	

Moreover, **SparkR**, **Vogoo**, **Duine**, **OpenSlopeOne**, **Apache Mahout**, **R** and **Giraph** resolve the problem of so-called collaborative filtering (Hosaka & Walet, 1997), which is supposed to extrapolate deduction about unavailable information on some instance A, in accordance to the available facts about the same problem of instance B, similar to the A one. For time series, **BigML** can be used, as it proves exponential smoothing, while it offers isolation forest for the anomaly detection. Additionally, so as to be capable of dealing with large amounts of data, **BigML** provides Latent Dirichlet Allocation for the problem of topic modeling. Moreover, a supplementary approach in data analysis is reasoning in the spectral domain, precisely the pattern mining, which is considered by **Apache Mahout** and **MOA**. Furthermore, another approach for pattern recognition could be optimizing predefined criterion function, which is why **Apache Mahout** and **VW** implements different optimization algorithms. Finally, when statistical methods are concerned, **Rhipe** can be used.

3.6 Big Data Visualization Tools

Classified as one of the ten V's of Big Data, and coming at the end of the Big Data processing pipeline after storage, processing and analysis, visualization plays a key role in knowledge representation. Without proper information presentation to the end user, the entire process would lose all purpose and so authors characterize this part as turning data mining into gold mining.

Figure 6. Examples of JavaScript visualization tools, source1, source2, source3

Information visualization has an important duty in large data set exploration and explanation because it represents the manner in which humans generally receive information. Some even go as far to state that, because the close relation that the visualization process has with the user, it can even be considered as important as the analytic process itself. Related literature (Caldarola & Rinaldi, 2017) explores the following aspects of visualization systems:

- **Scope**
- **Software category**
- **Visualization structure**
- **Operating system**
- **Licensing**
- **Scalability**
- **Extendibility**
- **Latest release version date**

Stating with open source (free) solutions that require no licensing as probably the most popular, JavaScript libraries called **Chart.js, Leaflet, Chartist.js, n3-charts, Sigma JS, Polymaps, Processing.js** and **Dyagraph** must be mentioned. Being programming libraries, these tools were primarily designed to be used by developer so they should not be considered as final presentation applications. Out of the mentioned set, and as their name might suggest in most cases, **Fusion Charts, Chart.js, Chartist.js, n3-charts,** and **Canvas** are made to work with charts as their primary visualization structure while **Leaflet** and **Polymaps** work with maps. On the other hand,

Processing.js handles images and **Sigma JS** graphs and networks. **Fusion Charts** is also a web-based JavaScript library handling charts, however its licensing is commercial.

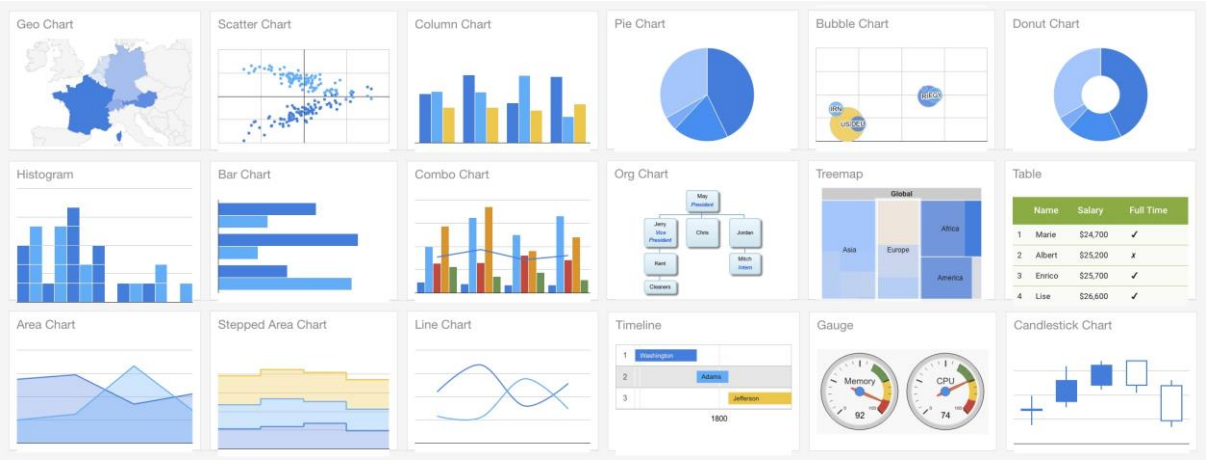

Figure 7. Google charts interface, source

On this line of web-based open source solutions, **Timeline JS** deserves to be mentioned as a web application handling timelines and designed for developers. **D3.js, Ember-charts** and **Google**

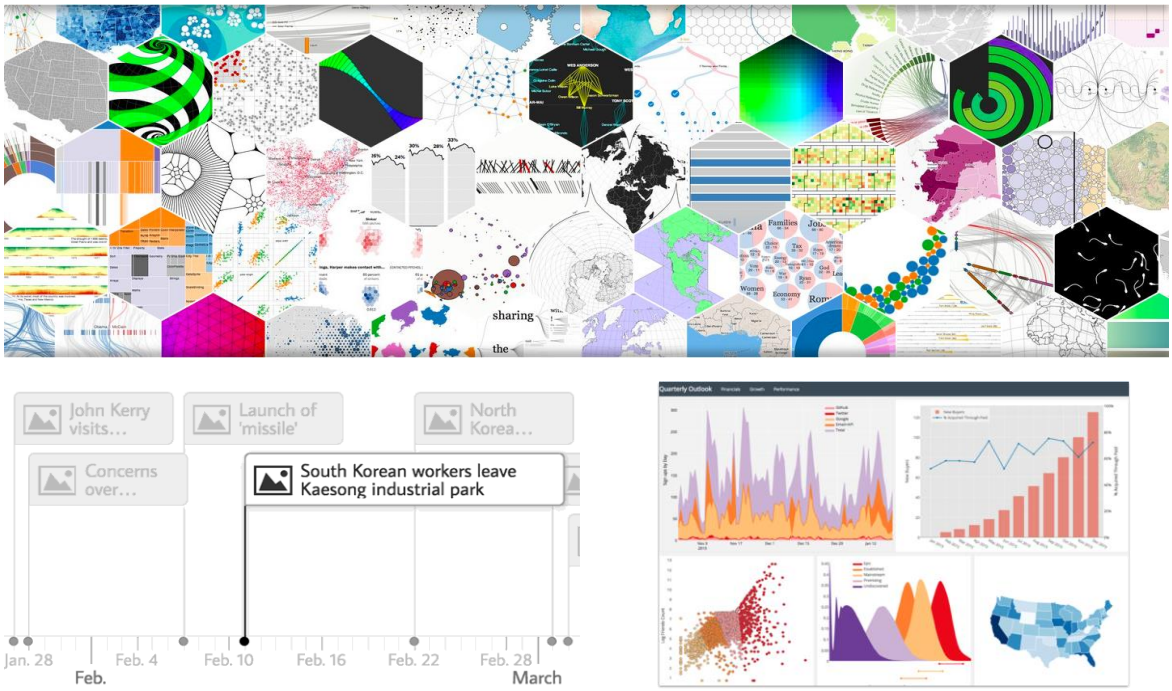

Figure 8. D3.js, TimelineJS and Plotly, source1, source2, source3

charts also offer JavaScript open-source web-based solutions including cloud scalability for developers with **D3.js** having the ability to work with charts, plots and maps simultaneously, **Ember-charts** and **Google charts** both offering charts, but **Google charts** supplementing this with tree maps, timelines, gauges etc. **Plotly**, on the other hand, requires commercial of community licensing, but offers both presentation and developer level web-based tools and a JavaScript or Python library handling charts, plots and maps.

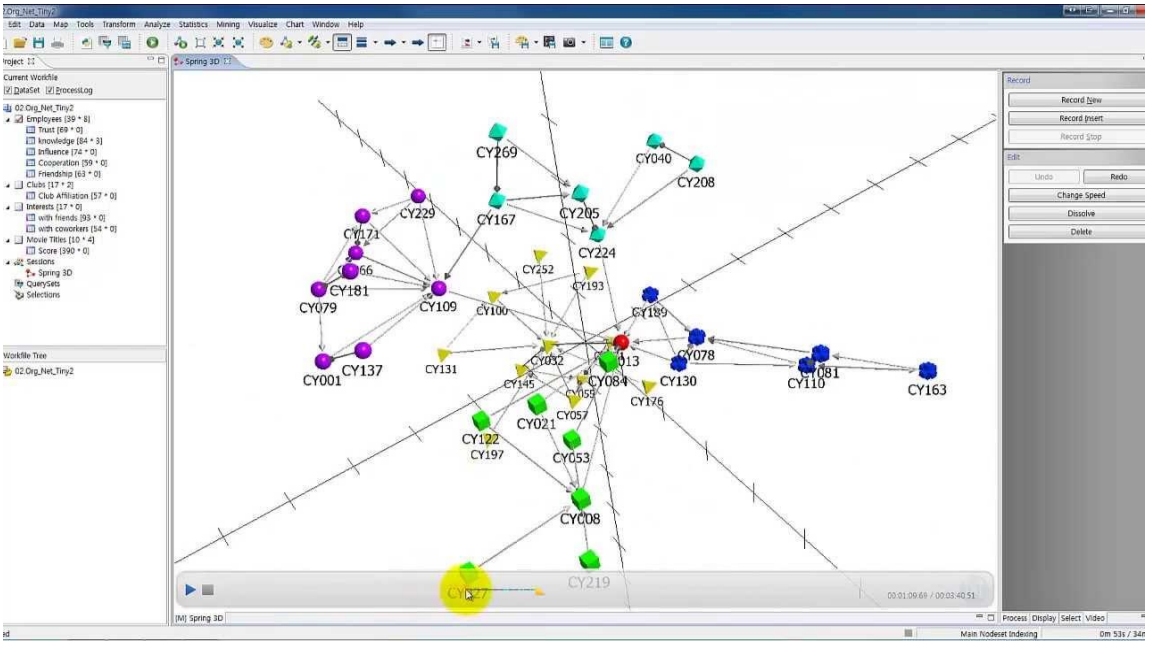

Figure 9. NetMiner, source

Open source software is also developed for non-web applications. **Cuttlefish, Cytoscape** and **Gephi** offer presentation level software frameworks for graphs and networks. **Cuttlefish** is multiplatform in nature because of its JVM base, with the other two supporting all three major operating systems: Windows, Linux and Mac OS. A presentation level desktop application with same multiplatform nature and visualization structure as **Cuttlefish** is **Graphwiz**. **Graph-tool** offers a Python module and is, therefore, developer usage oriented and can be implemented in all three operating systems. **JUNG,** on the other hand, is a JVM based multiplatform Java library also working with both graphs and networks. Commercial solutions for presentation with that visualization structure are offered by **Keynetiq, Netlytic,**

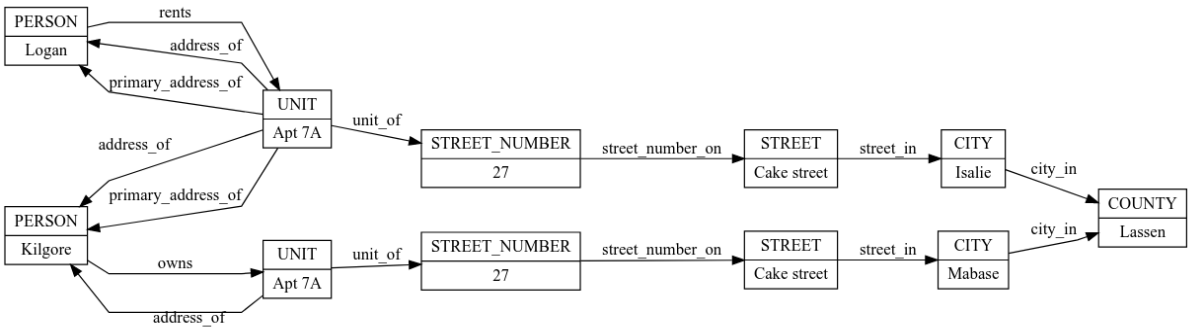

Figure 10. GraphWiz, souce1, source2

NetMiner with **Network Workbench** distinguishing itself with its ability to work with semantic network as opposed to the graph and network workload supported by the other three**.** Out of these four, the former two are software frameworks and the latter two desktop applications for windows. Some other cross-platform presentation tools worth mentioning are **NodeXL, Pajek, SocNetV, Sentinel Visualizer, Statnet, Tulip** and **Visone.** Out of the mentioned set, only **Sentinel Visualizer** and **Visione** are commercial, whilst others are free with some even being open-source. **Visone** offers free licensing to academic users. **NodeXL** is a template for Microsoft Excel whilst the others are Desktop applications with the exception of **Tulip,** an engine for relational data visualization being the only software framework. Other tools handle the usual combination of graphs and networks with **Sentinel Visualizer** featuring charts and 3D networks also.

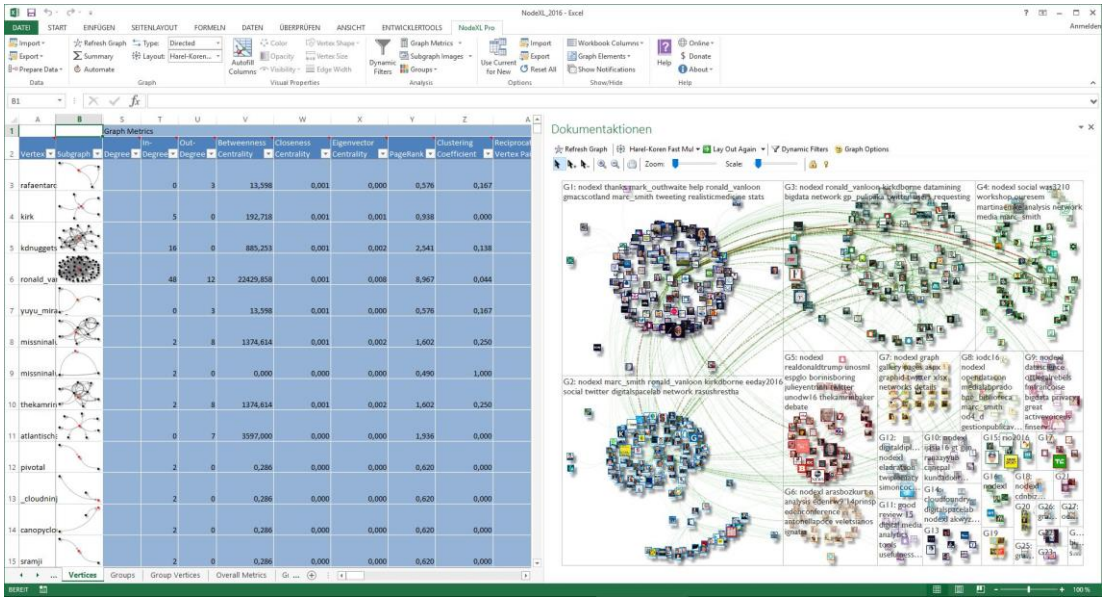

Figure 11. NodeXL, source

Finally, commercial presentation solutions **Tableau** and **Infogram** offer desktop applications and cloud hosting with support beyond chart, graph and map manipulation for Big Data applications with **Infogram** having native support for image and video processing also. Also falling in this commercial presentation category is the JavaScript **ChartBlocks** library.

Related papers (Ruan, Miao, Pan, Patterson, & Zhang, 2017) and (Genender-Feltheimer, 2018) offer concrete analysis on the issue of Big Data visualization.

Figure 12. Tableau and Infogram, source1, source2

4. Big Data Challenges

4.1 Challenges of Data

New era stated as Big Data was gendered for the fact that massive amounts of data are generated rapidly at every instance of time. Hence, approaches for its acquisition, storing, processing and deduction come across significant problems which were unknown to previously used data acquisition and processing concepts. Therefore, in this section, principal challenges for all aspects of extensive volume of data handling will be presented, having in mind previously explained V's, and accordingly to (Desai, 2018), (Katal, Wazid, & Goudar, 2013):

- **Heterogeneity** - due to the different data sources, acquired data structures varies significantly. Strictly, as stated in literature (Oussous, Benjelloun, Lahcen, & Belfkih, 2018), collected raw data can be in structured, semi-structured or unstructured format depending on its origination, which undoubtedly complicates its processing. Hence, the semi-structured or unstructured data, such as text documents, pictures, audio records, social network data etc, inevitable requires some pre-processing methods.
- **Uncertainty of data** - as for the coexistence of the various of data sources, the data reliability fluctuates fairly, meaning that numerous of missed, partial or even faulty measurements are often obtained, implying serious potential troubles when the data is to be analysed, once more indicating the necessity for firstly performing data managing, such as cleaning, filtering, transforming etc. (Wadhwani & Wang, 2017). Particularly, having inadequate or wrong data is taken as one of the most critical challenges, owing to the potential incorrect analysis output as a result of biased data.
- **Scalability** - one of most often highlighted bottlenecks in the context of Big Data. Namely, with time flowing, the amount of data is dramatically increasing. Hence, the scalability, as the capability of performing processing analysis on rapidly growing chunks of data, can be taken as one of the most relevant features that Big Data algorithms is supposed to contain. As stated in (Desai, 2018), only the incremental algorithms are resistant to this kind of data expansion. Furthermore, alike scalability, as a consequence of increasing amount of data, storage managing is also unavoidable barrier when Big Data regarded. Namely, this enormous quantity of acquired raw data, which is supposed to processed and analysed, requires gigantic memory capacity, bringing to the conclusion that the traditional depository solutions are insufficient for the considered zettabytes of data. Even clouds are regarded as inconvenient for the real-time applications, particularly because of the upload duration.
- **Timeliness** - considering real-time applications such as the stock market, financial fraud detection and transactions parsing, traffic management, energy optimization and so on, the output is required to be proceeded practically immediately. Precisely, the delay of the provided information is not allowed in any case, as for the fact that with the mentioned postponement, the supplied information might be completely useless. For example, if the money fraud case is to be considered, it is evident that information of the scam trying to be performed can be only effective just at the very same moment when it started.
- **Fault tolerance** - it is needless to explain the importance of the presented challenge of correctness, as it is supposed to be the main goal of any proposed solutions. Nonetheless, contrary to the traditional SQL approach, due to the high volume of data, its unstructured form, distributed nature and the necessity of near-to-real-time output response, there is no method that can fully guarantee on its fault tolerance.
- **Data security** - one of the most relevant problems in practise when the applications using private data are considered is its protection. It is obvious that in cases when, for example, weather forecasting or public transport management are deliberated, it can be taken as entirely irrelevant if it has been lost in some cases or publicly accessed. Nevertheless, if the healthcare, government or financial applications are to be considered, it is the highest

priority to ensure the data protection in both processing and storing phase of gathering information.

- **Visualization** - even though it does not affect the data processing anyhow, visualization is stated in literature as one of the crucial factors for Big Data Analytics. It cannot be surprising that without adequate results presentation, the derived knowledge can be considered worthless. Thus, the visualization tools will be introduced and discussed in details later in this deliverable.

Table 5. Big Data challenges and resolutions

	Storing	Processing	Analytics	Visualization
Heterogeneity	+	+		
Uncertainty of data		+	+	
Scalability	+	+	+	
Timeliness	+	+	+	
Fault tolerance		+	+	
Data security	+	+		
Visualization				+

In order to overcome these challenges, many platforms were developed and they will be presented and extensively discussed in the next section. Nonetheless, all of them are organized in a specific way, precisely designed so as to cover four main steps - **storing**, **processing**, **analytics** and **visualization**. Therefore, in Table 5 the specified challenges and the appropriate resolutions are presented. Accordingly, to the previously stated heterogeneity and data security definition, it is undeniable that storing and processing are responsible for their handling. Furthermore, besides them, for scalability and timeliness are also influenced by the selected analytics approach, as for its potential scalability barrier or the inability for real-time output delivery. Additionally, the data uncertainty and fault tolerance are to be managed by proper choice of the processing and analytics tool.

4.2 Challenges in Big Data Landscape

There are a plethora of technologies and this wide range of technologies bring another layer of challenges briefly mentioned in this section.

Big Data Technologies

In most of the cases, there is usually more than one tool fitting a particular scenario and it is challenging to figure out the optimal solution to the given problem. There are multiple objectives associated with such choices like cost, environment, integration, compatibility, expertise, making it hard to make the right choice. This can be easy to get lost in the variety of big data technologies now available on the market. It is challenging to figure if the organization needs Spark or would the speeds of Hadoop MapReduce be enough? Is it better to store data in Cassandra or HBase? Finding the right answers can be tricky. It is easier to choose poorly, given the ocean of technological opportunities without a clear view exact needs beforehand.

Uncertainty in Data Management

The most obvious challenge associated with big data is simply storing and analyzing all the information. The problem with big data management is the range of tools, there are SQL frameworks and NoSQL frameworks and a variety of approaches such as hierarchical object representation (such as JSON, XML and BSON) and the concept of key-value storage. Much of the additional data is unstructured, meaning that it doesn't reside in a database. Documents, photos, audio, videos and other unstructured data can be difficult to search and analyze. The landscape of tools and frameworks in plummeting but there are no rules of thumb to deal with the heterogeneous data, storage, searching and integration of these tools. The wide range of tools, developers and the status of the market are creating uncertainty with the data management.

Talent Gap

There appears to be a scarce understanding of the big data tools and frameworks. It is either hard or too costly to find the right experts to do the task and sometimes even harder to narrow down the exact big data needs and requirements for the task. There is a lack of experienced people and certified data scientists available at present. Training people at entry level can be expensive for a company dealing with new technologies. Many are instead working on automation solutions involving Machine Learning and Artificial Intelligence to build insights, but this also takes a well-trained staff or the outsourcing of skilled developers.

Cloud or Premises

A major challenge is to choose between building the cloud on premises or using the platform as a service. This offers a tradeoff between cost and long-term benefits with expert availability. If an organization opts for an on-premises solution, there are costs of new hardware, new hires (administrators and developers), electricity etc. Although the needed frameworks can be open-source, experts are needed for the development, setup, configuration and maintenance of new software.

If an organization decides on a cloud-based big data solution, it would still need experts for big data solution development and setup and maintenance of needed frameworks, in addition to payment for cloud services. In both cases, the need for future expansions is always desired.

Scalability

With big data, it’s crucial to be able to scale up and down on-demand. Many organizations fail to take into account how quickly a big data project can grow and evolve. Constantly pausing a project to add additional resources cuts into time for data analysis. Big data workloads also tend to be bursty, making it difficult to predict where resources should be allocated. The extent of this big data challenge varies by the solution. A solution in the cloud will scale much easier and faster than an on-premises solution

Choosing the Cloud service

Many big data players are gradually building open platforms (PAAS) through a home built products and implementation of popular open source engines and are increasingly getting closer to one-stop-shop for many enterprises. However, it is not easy to choose the vendor or services to use.

For example, AWS offers analytics frameworks, real-time analytics, databases business intelligence, AI, and deep learning capabilities. Google offers Big Data (BigQuery, Dataflow, Dataproc, etc.) and it has a lot to offer in AI, ranging from including multilingual translation to using a new machine learning API for video recognition. The larger enterprise IT vendors like Microsoft, IBM, SAP, Oracle and Salesforce also offer big data and AI solutions, both in the cloud (most noticeably, Microsoft) and on premises.

Data Quality

The data being gathered from all the possible sources can pose a range of quality problems. There could be the inaccurate customer or contact data, Large amount of data make it hard to focus on a particular aspect of data, or difficult to find the optimal aspect of the data to focus upon. There can be a problem of duplicate data or obsolete data. There could be problems with data compliance issues. The heterogeneity of data makes it even harder to apply unified or standardized quality measures to assess the data quality.

Organizational Resistance

Sometimes the member of the organizations can also resist the change and embracement of the new technologies. In the NewVantage Partners survey, 85.5 per cent of the people said that their organizations intended to create a data-driven culture, but only 37.1 had been successful with those efforts.

5. Summary

Recently acquired data form a variety of different sources is rapidly increasing, generating plentiful previously unknown problems in all distinctive spheres of applications such as social networks, public health, energy and finance management, education, logistic etc. Hence, numerous research has been conducted during previous years in order to improve and review current performances, so in this deliverable, the state of the art analysis on Big Data concept, challenges and innovative technologies is extensively discussed.

In an ever growing and quickly evolving community, new solutions seem to be proposed every so often and so it can be extremely difficult to capture all relevant technologies in a comprehensive way. The shear velocity at which this segment of the industry is evolving is best described by Figure 13 where the Big Data landscape is depicted in 2012, 2016, 2017 and 2018.

Due to the complex characteristics of Big Data described best by its V's, various challenges such as heterogeneity, storage management, scalability, security, visualization etc. were presented. Moreover, current proposed solutions for them, the preprocessing, streaming, analytics and visualization tools were discussed and compared. Nonetheless, a deluge of barriers is still existing, so the possibilities for their improvements is not just desired, but essential.

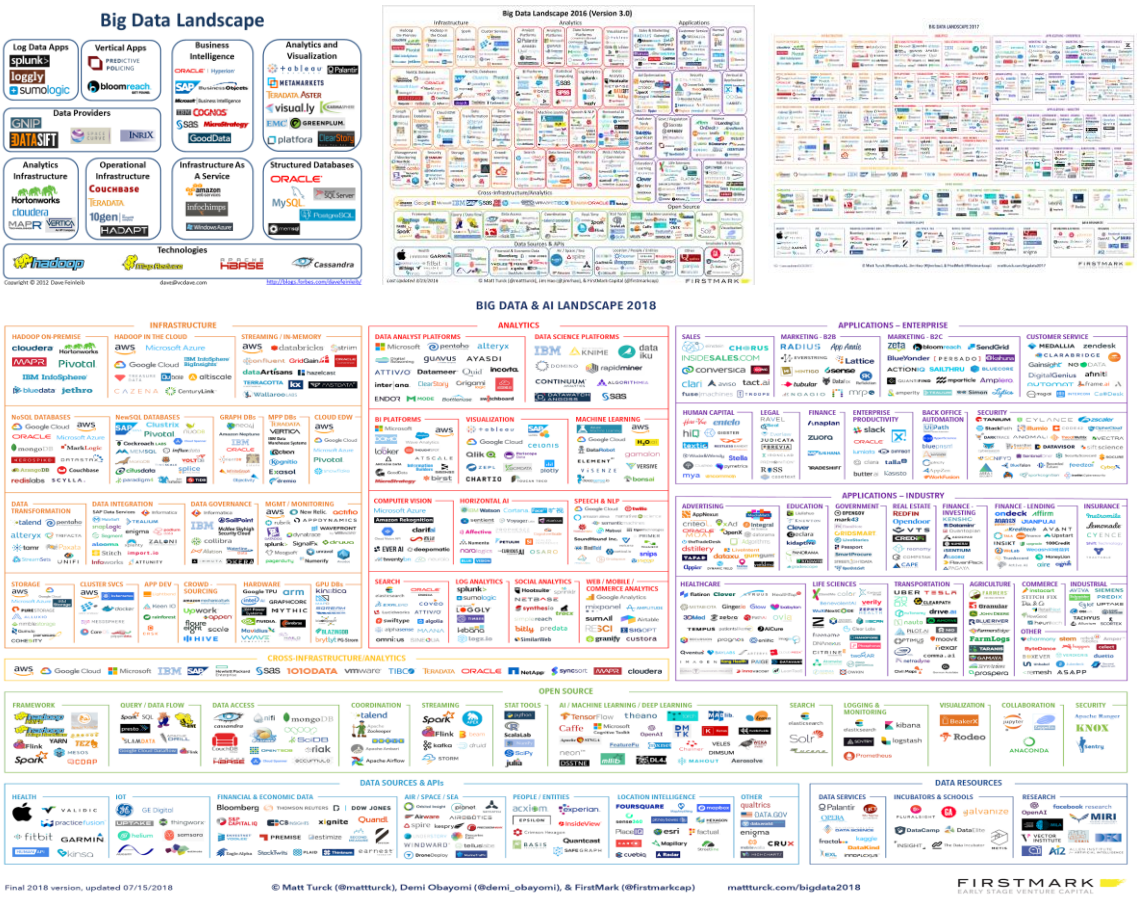

Figure 13. The evolution of the Big Data landscape, source1, source2, source3, source4

References

Ahmed, E., Yaqoob, I., Hashem, I. A., Khan, I., Ahmed, A. I., Imran, M., & Vasilakos, A. (2017). The role of big data analytics in Internet of Things. *Computer Networks*, 459-471.

Ali, M., Gao, F., & Mileo, A. (2015). CityBench: A Configurable Benchmark to Evaluate RSP Engines using Smart City Datasets. *ISWC 2015: The Semantic Web - ISWC 2015*, (pp. 374-389).

Amin, A., Al-Obeidat, F., Shah, B., Adnan, A., Loo, J., & Anwar, S. (2019). Customer churn prediction in telecommunication industry using data certainty. *Journal of Business Research*, 290-301.

Anicic, D., Fodor, P., Rudolph, S., & Stojanovic, N. (2011). EP-SPARQL: A Unified Language for Event Processing and Stream Reasoning. *20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011.* Hyderabad.

Aron, J., Niemann, B., & . (2014). Sharing best practices for the implementation of Big Data applications in government and science communities. *2014 IEEE International Conference on Big Data (Big Data).* Washington, DC, USA: IEEE.

Barbieri, D., Braga, D., Ceri, S., Valle, E., & Grossniklaus, M. (2010). C-SPARQL: A CONTINUOUS QUERY LANGUAGE FOR RDF DATA STREAMS. *International Journal of Semantic Computing*, 3-25.

Bataev, A. V. (2018). Analysis of the Application of Big Data Technologies in the Financial Sphere. *2018 IEEE International Conference "Quality Management, Transport and Information Security, Information Technologies" (IT&QM&IS).* St. Petersburg, Russia: IEEE.

Bayne, L. E. (2018). Big Data in Neonatal Health Care: Big Reach, Big Reward? *Critical Care Nursing Clinics of North America*, 481-497.

Begenau, J., Farboodi, M., & Veldkamp, L. (2018). Big data in finance and the growth of large firms. *Journal of Monetary Economics*, 71-87.

Bellomarini, L., Gottlob, G., & Sallinger, E. (2018). The Vadalog System: Datalog-based Reasoning for Knowledge Graphs. *arXiv:1807.08709*.

Bharat, V., Shubham, S., & Jagdish, D. (2017). Smart water management system in cities. *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC).* Chirala, India: IEEE.

Bolles, A., Grawunder, M., & Jacobi, J. (2008). Streaming SPARQL - Extending SPARQL to Process Data Streams. *ESWC 2008: The Semantic Web: Research and Applications*, (pp. 448-462).

Bozzon, A., Cudre-Maroux, P., & Pautasso, C. (2016). Web Engineering. *16th International Conference, ICWE 2016.* Lugano.

Caldarola, E. G., & Rinaldi, A. M. (2017). Big Data Visualization Tools: A Survey - The New Paradigms, Methodologies and Tools for Large Data Sets Visualization. *6th International Conference on Data Science, Technology and Applications (DATA 2017).*

Chauhan, J., Chowdhury, S., & Makaroff, D. (2012). Performance Evaluation of Yahoo! S4: A First Look. *2012 7th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing.*

Corbellini, A., Mateos, C., Zunino, A., Godoy, D., & Schiaffino, S. (2017). Persisting big-data: The NoSQL landscape. *Information Systems*, 1-23.

Das, N., Rautaray, S., & Pandey, M. (2018). Big Data Analytics for Medical Applications. *International Journal of Modern Education and Computer Science*.

Dean, J., & Ghemawat, S. (2014). MapReduce: Simplified Data Processing on Large Clusters. *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6.*

Dellachiesa, M., Ebaid, A., Eldway, A., Elmagarmid, A., Ilyas, I. F., Ouzzani, M., & Tang, N. (2013). NADEEF: A Commodity Data Cleaning System. *2013 ACM SIGMOD International Conference on Management of Data.*

Dell'Agilo, D., Calbimonte, J., Balduini, M., Corcho, O., & Della Valle, E. (2013). On Correctness in RDF Stream Processor Benchmarking. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface.*

Dell'Aglio, D., Balduini, M., & Della Valle, E. (n.d.). On the need to include functional testing in RDF stream engine benchmarks.

Desai, P. V. (2018). A survey on big data applications and challenges. *Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018).* IEEE.

Dey, A. (2016). Machine Learning Algorithms: A Review. *International Journal of Computer Science and Information Technologies*, 1174-1179.

Doug, L. (2011). 3D Data Management: Controlling Data Volume, Velocity and Variety. *Application Delivery Strategies*.

Friedman, E., & Tzoumas, K. (2016). *Introduction to Apache Flink: Stream Processing for Real Time and Beyond.* O'Reilly Media.

Garcia, S., Ramirez-Gallego, S., Luengo, J., Benitez, J., & Herrera, F. (2016). Big data preprocessing: methods and prospects. *Big Data Analytics*.

Ge, M., Bangui, H., & Buhnova, B. (2018). Big Data for Internet of Things: A Survey. *Future Generation Computer Systems*, 601-614.

Genender-Feltheimer, A. (2018). Visualizing High Dimensional and Big Data. *Procedia Computer Science*, 112-121.

Ghani, N. A., Hamid, S., Hashem, I. A., & Ahmed, E. (2018). Social media big data analytics: A survey. *Computers in Human Behavior*.

Ghemawat, S., Gobioff, H., & Leung, S.-T. (2003). The Google File System. *Proceedings of the 19th ACM Symposium on Operating Systems Principles.* Bolton Landing, NY, USA.

Ghofrani, F., He, Q., Goverde, R., & Liu, X. (2018). Recent applications of big data analytics in railway transportation systems: A survey. *Transportation Research Part C: Emerging Technologies*, 226-246.

Gohar, M., Muzammal, M., & Ur Rahman, A. (2018). SMART TSS: Defining transportation system behavior using big data analytics in smart cities. *Sustainable Cities and Society*, 114-119.

Grant-Muller, S., Hodgson, F., & Malleson, N. (2017). Enhancing Energy, Health and Security Policy by Extracting, Enriching and Interfacing Next Generation Data in the Transport Domain (A Study on the Use of Big Data in Cross-Sectoral Policy Development). *2017 IEEE International Congress on Big Data (BigData Congress).* Honolulu, HI, USA: IEEE.

Gutierrez, D. D. (2017). *Big Data for Finance.* Technical report.

Hardy, K., & Maurushat, A. (2017). Opening up government data for Big Data analysis and public benefit. *Computer Law & Security Review*, 30-37.

Hasan, A., Kalıpsız, O., & Akyokuş, S. (2017). Predicting financial market in big data: Deep learning. *2017 International Conference on Computer Science and Engineering (UBMK).* Antalya, Turkey: IEEE.

Hathaway, R., & Bezdek, J. (2002). Clustering incomplete relational data using the non-Euclidean relational fuzzy c-means algorithm. *Pattern Recognition Letters*, 151-160.

He, Y., Yu, F. R., & Zhao, N. (2016). Big Data Analytics in Mobile Cellular Networks. *IEEE Access*, 1985-1996.

Hosaka, A., & Walet, N. (1997). Algebraic method for large-Nc QCD. *Australian Journal of Physics*, 211-220.

Huang, W.-Y., Chen, A.-P., & Hsu, Y.-H. (2016). Applying Market Profile Theory to Analyze Financial Big Data and Discover Financial Market Trading Behavior - A Case Study of Taiwan Futures Market. *2016 7th International Conference on Cloud Computing and Big Data (CCBD).* Macau, China: IEEE.

Imawan, A., & Kwon, J. (2015). A timeline visualization system for road traffic big data. *2015 IEEE International Conference on Big Data (Big Data).* Santa Clara, CA, USA: IEEE.

Inoubli, W., Aridhi, S., Mezni, H., Maddouri, M., & Nguifo, E. (2018). An experimental survey on big data frameworks. *Future Generation Computer Systems*, 546-564.

Istepanian, R. S., & Al-Anzi, T. (2018). m-Health 2.0: New perspectives on mobile health, machine learning and big data analytics. *Methods*, 34-40.

Jun, C., & Chung, C. (2016). Big data analysis of local government 3.0: Focusing on Gyeongsangbuk-do in Korea. *Technological Forecasting and Social Change*, 3-12.

Kalbandi, I., & Anuradha, J. (2015). A Brief Introduction on Big Data 5Vs Characteristics and Hadoop Technology. *Procedia Computer Science*.

Katal, A., Wazid, M., & Goudar, R. H. (2013). Big data: Issues, challenges, tools and Good practices. *2013 Sixth International Conference on Contemporary Computing (IC3).* Noida, India: IEEE.

Kaul, L., & Goudar, R. H. (2016). Internet of things and Big Data - challenges. *2016 Online International Conference on Green Engineering and Technologies (IC-GET).* Coimbatore, India: IEEE.

Khan, M. A.-u.-d., Uddin, M. F., & Gupta, N. (2014). Seven V's of Big Data understanding Big Data to extract value. *Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education.* Bridgeport, CT, USA: IEEE.

Khan, N., Alsaqer, M., Shah, H., Badsha, G., Abbasi, A. A., & Salehian, S. (2018). The 10 Vs, Issues and Challenges of Big Data. In *ICBDE.*

Kharlamov, E. (2017). Semantic access to streaming and static data at Siemens. *Journal of Web Semantics*, 54-74.

Khayyat, Z., Illyas, I. F., Jindal, A., Madden, S., Ouzzani, M., Papotti, P., . . . Yin, S. (2015). BigDansing: A System for Big Data Cleansing. *2015 ACM SIGMOD International Conference on Management of Data*, (pp. 1215-1230).

Kobusińska, A., Leung, C., Hsu, C.-H., Raghavendra, S., & Chang, V. (2018). Emerging trends, issues and challenges in Internet of Things, Big Data and cloud computing. *Future Generation Computer Systems*, 416-419.

Kolb, L., Thor, A., & Rahm, E. (2012). Dedoop: efficient deduplication with Hadoop. *VLBD Endowment.*

Koo, D., Piratla, K., & Matthews, J. C. (2015). Towards Sustainable Water Supply: Schematic Development of Big Data Collection Using Internet of Things (IoT). *Procedia Engineering*, 489-497.

Kreps, J. (2011). Kafka : a Distributed Messaging System for Log Processing.

Ku, T.-Y., Park, W.-K., & Choi, H. (2017). IoT energy management platform for microgrid. *2017 IEEE 7th International Conference on Power and Energy Systems (ICPES).* Toronto, ON, Canada: IEEE.

Kumari, A., Tanwar, S., Tyagi, S., Kumar, N., Maasberg, M., & Choo, K.-K. R. (2018). Multimedia big data computing and Internet of Things applications: A taxonomy and process model. *Journal of Network and Computer Applications*, 169-195.

Lachhab, F., Bakhouya, M., Ouladsine, R., & Essaadi, M. (2016). Performance evaluation of CEP engines for stream data processing. *2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech).*

Lakshen, G. A., Vraneš, S., & Janev, V. (2016). Big data and quality: A literature review. *2016 24th Telecommunications Forum (TELFOR).* Belgrade, Serbia: IEEE.

Le Phuoc, D., Dao-Tran, M., Xavier Parreira, J., & Hauswirth, M. (2011). A Native and Adaptive Approach for Unified Processing of Linked Streams and Linked Data. *The Semantic Web – ISWC 2011: 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011,*, (pp. 370-388). Bonn.

Leavitt, N. (2010). Will NoSQL Databases Live Up to Their Promise? *Computer*, 12-14.

Lee, C. H., & Yoon, H.-J. (2017). Medical big data: promise and challenges. In *Kidney research and clinical practice.*

Lehmann, J., Sejdiu, G., Bühmann, L., Westphal, P., Stadler, C., Ermilov, I., & Bin, S. (2017). Distributed Semantic Analytics Using the SANSA Stack. *The Semantic Web – ISWC 2017.*

Leng, Y. (2014). Incomplete Big Data Distributed Clustering. *Applied Mechanics and Materials*, 1496-1499.

Le-Phuoc, D., Dao-Tran, M., Pham, M., Boncz, P., Eiter, T., & Fink, M. (2012). Linked Stream Data Processing Engines: Facts and Figures. *ISWC 2012: The Semantic Web – ISWC 2012*, (pp. 300-312).

Li, J., & Huang, X. (2017). Target Customer Selection Method Based on Data Mining in Big Data Environment. *2017 International Conference on Smart Grid and Electrical Automation (ICSGEA).* Changsha, China: IEEE.

Li, Y., & Zhai, X. (2018). Review and Prospect of Modern Education using Big Data. *Procedia Computer Science*, 341-347.

Li, Y., Li, P., & Zhu, F. (2017). Design of higher education quality monitoring and evaluation platform based on big data. *2017 12th International Conference on Computer Science and Education (ICCSE).* Houston, TX, USA: IEEE.

Liang, J., Yang, J., & Wu, Y. (2016). Big Data Application in Education: Dropout Prediction in Edx MOOCs. *2016 IEEE Second International Conference on Multimedia Big Data (BigMM).* Taipei, Taiwan: IEEE.

Liu, D. (2018). Big Data Analytics Architecture for Internet-of-Vehicles Based on the Spark. *2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS).* IEEE: Xiamen, China.

Liu, S., McGree, J., Ge, Z., & Xie, Y. (2016). 8 - Big data from mobile devices. *Computational and Statistical Methods for Analysing Big Data with Applications*, 157-186.

Martis, R. J., Gurupur, V. P., Lin, H., Islam, A., & Fernandes, S. L. (2018). Recent advances in Big Data Analytics, Internet of Things and Machine Learning. *Future Generation Computer Systems*, 696-698.

Matsebula, F., & Mnkandla, E. (2017). A big data architecture for learning analytics in higher education. *2017 IEEE AFRICON.* Cape Town, South Africa: IEEE.

Munshi, A. A., & Mohamed, Y. (2017). Big data framework for analytics in smart grids. *Electric Power Systems Research*, 369-380.

Nelson, M. S., & Pouchard, L. (2017). A pilot “big data” education modular curriculum for engineering graduate education: Development and implementation. *2017 IEEE Frontiers in Education Conference (FIE).* Indianapolis, IN, USA: IEEE.

Olayinka, O., Kekeh, M., Sheth-Chandra, M., & Akpinar-Elci, M. (2017). Big Data Knowledge in Global Health Education. *Annals of Global Health*, 676-681.

Olston, C., Reed, B., Srivastava, U., Kumar, R., & Tomkins, A. (2008). Pig Latin: A Not-So-Foreign Language for Data Processing. *Proceedings of the 2008 ACM SIGMOD international conference on Management of data.*

Óskarsdóttir, M., Bravo, C., Sarraute, C., Vanthienen, J., & Baesens, B. (2019). The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics. *Applied Soft Computing*, 26-39.

Oussous, A., Banjelloun, F., Lahcen, A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*, 431-448.

Oussous, A., Benjelloun, F.-Z., Lahcen, A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*, 431-448.

Panimalar, A., Shree, V., & Kathrine, V. (2017). The 17 V’s Of Big Data. *International Research Journal of Engineering and Technology (IRJET)*.

Pardos, Z. (2017). Big data in education and the models that love them. *Current Opinion in Behavioral Sciences*, 107-113.

Patel, J. (2016). Operational NoSQL Systems: What's New and What's Next? *Computer*, 23-30.

Peng, S., Yu, S., & Mueller, P. (2018). Social networking big data: Opportunities, solutions, and challenges. *Future Generation Computer Systems*, 1456-1458.

Persico, V., Pescapé, A., Picariello, A., & Sperlí, G. (2018). Benchmarking big data architectures for social networks data processing using public cloud platforms. *Future Generation Computer Systems*, 98-109.

Phan, S., & Chen, C. (2017). 9 - Big Data and Monitoring the Grid. *The Power Grid*, 253-285.

Prakash, M., Padmapriy, G., & Kumar, M. (2018). A Review on Machine Learning Big Data using R. *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT).*

Qiu, R., Huang, Z., & Patel, I. (2015). A big data approach to assessing the US higher education service. *2015 12th International Conference on Service Systems and Service Management (ICSSSM).* Guangzhou, China: IEEE.

Rai, R., & Chettri, P. (2018). Chapter Six - NoSQL Hands On. *Advances in Computers*, 157-277.

Ren, X., Khrouf, H., Kazi-Aoul, Z., Chabchoub, Y., & Cure, O. (2016). On measuring performances of C-SPARQL and CQELS.

Rinne, M., Nuutila, E., & Torma, S. (2012). INSTANS: High-Performance Event Processing with Standard RDF and SPARQL. *Poster session of the 11th International Semantic Web Conference (ISWC 2012), At Boston, USA.* Boston.

Ristevski, B., & Chen, M. (2018). Big Data Analytics in Medicine and Healthcare. *Journal of Integrative Bioinformatics*.

Ruan, Z., Miao, Y., Pan, L., Patterson, N., & Zhang, J. (2017). Visualization of big data security: a case study on the KDD99 cup data set. *Digital Communications and Networks*, 250-259.

Saggi, M., & Jain, S. (2018). A survey towards an integration of big data analytics to big insights for value-creation. *Information Processing and Management*, 758-790.

Saleh, I., Tan, W., & Blake, M. (2013). Social-Network-Sourced Big Data Analytics. *IEEE Internet Computing*.

Santoso, L. W., & Yulia. (2017). Data Warehouse with Big Data Technology for Higher Education. *Procedia Computer Science*, 93-99.

Schwerdtle, P., & Bonnamy, J. (2017). Big Data in Nurse Education. *Nurse Education Today*, 114-116.

Shah, F., Li, J., & Shah, Y. (2017). Broad big data domain via medical big data. *2017 4th International Conference on Systems and Informatics (ICSAI).* Hangzhou, China: IEEE.

Sharma, D., & Kumar, N. (2017). A Review on Machine Learning Algorithms, Tasks and Applications. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*.

Shi, Z., & Wang, G. (2018). Integration of big-data ERP and business analytics (BA). *The Journal of High Technology Management Research*, 141-150.

Shvachko, K., Kuang, H., & Radia, S. (2010). The Hadoop Distributed File System. *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST).* Incline Village, NV, USA.

Sixin, X., Yayuan, Y., & Jiang, Y. J. (2017). A New Governance Architecture for Government Information Resources Based on Big Data Ecological Environment in China. *2017 IEEE International Symposium on Multimedia (ISM).* Taichung, Taiwan: IEEE.

Sun, C., Gao, R., & Xi, H. (2014). Big data based retail recommender system of non E-commerce. *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT).* Hefei, China: IEEE.

Surshanov, S. (2015). Using apache storm for big data. *COMPUTER MODELLING & NEW TECHNOLOGIES 2015.*

Tafti, A. P., Behravesh, E., & Assefi, M. (2017). bigNN: An open-source big data toolkit focused on biomedical sentence classification. *2017 IEEE International Conference on Big Data (Big Data).* Boston, MA, USA: IEEE.

Tang, N. (2014). Big Data Cleaning. *Lecture Notes in Computer Science*, 13-24.

Tian, X., Han, R., Wang, L., Lu, G., & Zhan, J. (2015). Latency critical big data computing in finance. *The Journal of Finance and Data Science*.

Tomasini, R., Della Valle, E., Mauri, A., & Barambilla, M. (2017). RSPLab: RDF Stream Processing Benchmarking Made Easy. *ISWC 2017: The Semantic Web – ISWC 2017*, (pp. 202-209).

Torre-Bastida, A., Del Ser, J., & Laña, I. (2018). Big Data for transportation and mobility: recent advances, trends and challenges. *IET Intelligent Transport Systems*, 742-755.

Tsuge, S., Shishibori, M., Kuroiwa, S., & Kita, K. (2001). Dimensionality reduction using non-negative matrix factorization for information retrieval. *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236).*

Tu, C., He, X., Shuai, Z., & Jiang, F. (2017). Big data issues in smart grid – A review. *Renewable and Sustainable Energy Reviews*, 1099-1107.

Van, C., Gao, F., & Ali, M. (2017). Optimizing the Performance for Concurrent RDF Stream Processing Queries. *ESWC 2017: The Semantic Web*, (pp. 238-253).

Wadhwani, K., & Wang, Y. (2017). *Big Data Challenges and Solutions.* Technical report.

Wang, C., Li, X., Zhou, X., Wang, A., & Nedjah, N. (2016). Soft computing in big data intelligent transportation systems. *Applied Soft Computing*, 1099-1108.

Wang, L., & Ranjan, R. (2015). Processing Distributed Internet of Things Data in Clouds. *IEEE Cloud Computing*, 76-80.

Wu, P.-J., & Chen, Y.-C. (2017). Big data analytics for transport systems to achieve environmental sustainability. *2017 International Conference on Applied System Innovation (ICASI).* Sapporo, Japan: IEEE.

Wu, P.-J., & Lin, K.-C. (2018). Unstructured big data analytics for retrieving e-commerce logistics knowledge. *Telematics and Informatics*, 237-244.

Yan, Z. (2018). Big data and government governance. *2018 International Conference on Information Management and Processing (ICIMP).* London, UK: IEEE.

Yazti, D. Z., & Krishnaswamy, S. (2014). Mobile Big Data Analytics: Research, Practice, and Opportunities. *2014 IEEE 15th International Conference on Mobile Data Management.* Brisbane, QLD, Australia: IEEE.

Yu, S., Yang, D., & Feng, X. (2017). A Big Data Analysis Method for Online Education. *2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA).* Changsha, China: IEEE.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: cluster computing with working sets. *HotCloud'10 Proceedings of the 2nd USENIX conference on Hot topics in cloud computing.*

Zeyu, J., Shuiping, Y., Mingduan, Z., Yongqiang, C., & Yi, L. (2017). Model Study for Intelligent Transportation System with Big Data. *Procedia Computer Science*, 418-426.

Zhang, P., Xiong, F., Gao, J., & Wang, J. (2017). Data Quality in Big Data Processing: Issues, Solutions and Open Problems. *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI).*

Zhang, S., Jia, S., & Ma, C. (2018). Impacts of public transportation fare reduction policy on urban public transport sharing rate based on big data analysis. *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA).* Chengdu, China: IEEE.

Zhang, Y., Duc, P., Corchno, O., & Calbimonte, J. (2012). SRBench: A Streaming RDF/SPARQL Benchmark. *ISWC 2012: The Semantic Web – ISWC 2012*, (pp. 641-657).

Zohrevand, Z., Glasser, U., & Shahir, H. Y. (2016). Hidden Markov based anomaly detection for water supply systems. *2016 IEEE International Conference on Big Data (Big Data).* Washington, DC, USA: IEEE.