A State-of-the-Art Review on Big Data Technologies

Marko Jelić[*a], Dea Pujić[*a], Dejan Paunović[*], Hajira Jabeen[**]
[*] The Mihajlo Pupin Institute, University of Belgrade, Belgrade, Serbia
[a] School of Electrical Engineering, University of Belgrade, Belgrade, Serbia
[**] Computer Science Institute, University of Bonn, Bonn, Germany
{marko.jelic, dea.pujic, dejan.paunovic}@pupin.rs, jabeen@iai.uni-bonn.de

***Abstract* — As the level of implementation and integration of digital systems in every part of daily life constantly increases, higher and higher volumes of data are being generated at unprecedented rates without any indication of slowing down at any point in the near future. The contemporary IT infrastructure has started to struggle with the storage, processing, analysis and knowledge extraction capacities required for maintaining such large amounts of data, therefore various novel solutions have been developed in order to deal with this new phenomenon, named Big data. With the number of related solutions ever growing, a requirement arose for a systematization of methods and technologies related to this domain. Therefore, this paper aims to provide an overview of the current state-of-the-art solutions related to Big data.**

I. INTRODUCTION

The term Big data has gained a lot of momentum in recent years, however even though it is used more and more within the scientific community for a wide variety of different applications, there is no single definition of what Big data actually represents. Nonetheless, Big data generally refers to datasets with high volume of the order of magnitude of exabytes (10^{18} B) and greater. As analogue technologies are slowly replaced with their respective digital counterparts, the data logs associated with these new solutions often generate high amounts of data comparable to the aforementioned limit on a daily basis, especially within large infrastructures and big corporations. With data mining, machine learning and data sciences in general being probably the most researched topics of computer science currently, the hunger for data is ever-growing. When working with associated enormous data amounts, the modern-day IT infrastructure and establishment find themselves in a situation where the processing power and storage capabilities are simply too limited for the task required. Therefore, novel solutions have started to appear in the domain of Big data storage, management, processing, analytics and visualization.

II. CHARACTERIZATION OF BIG DATA

The core characteristics of Big data were originally defined by the so-called 3V's of Big data [1], [2] being: *volume* – the large amount of data that has to be captured, stored, processed and displayed, *velocity* – the rate at which the data is being generated, or analyzed within the system and *variety* – the differences in structure of the incoming data and the differences in data sources themselves.

As the technology progressed, more V's of Big data were singled out to describe new challenges that were faced in this domain giving us lists of 5V's [3] and 7V's [4] of Big data: *veracity* – the truthfulness and uncertainty of data, the concept of equally represented classes of data, employing correct methodologies for data processing and analytics etc., *validity* – the suitability of the selected data for the chosen application, *volatility* – the temporal validity and fluency of the data and *value* – the (useful) information extracted from the data, sometimes regarded as the most important characteristic of Big data. Some papers [5] also go a step further and include *visualization* (the question of properly displaying and showcasing the information derived from Big data processing), *vulnerability* (the security issues associated with Big data) and v*ariability* (the inconsistency i.e. the changing meaning of the data) to form the 10V's with some [6] even going as far as to include 17V's to describe the nature of Big data. Regardless of how many descriptors are isolated when describing the nature of Big data, it is abundantly clear that the nature of Big data is highly complex and that as such requires special technical solutions for every step in the data workflow. This paper aims to summarize the current state of technologies in the Big data domain.

III. BIG DATA WORKFLOW

Having in mind that reliance of modern-day technologies on data is ever increasing, it is no wonder that capabilities of working with Big Data are often singled out as one of the most important features of these solutions. Therefore, different techniques which are employed in the process of its acquisition, storing, processing and information derivation have to tackle various problems when performing these tasks. This section aims at outlining the key challenges that relate to workflows with high amounts of data, specially having in mind the previously explained V's, and in accordance with [7], [8]:

- *Heterogeneity* - The differences in data sources produce significant inconsistencies with the acquired data structure. As stated in literature [9], raw data structure format is classified as either structured, semi-structured or unstructured in accordance with its origin. Therefore, workflows with semi-structured and unstructured data (e.g. text documents, pictures, audio recordings, social network data, etc.) must include a preprocessing module in some form.

- *Uncertainty of data* - When simultaneously working with different data sources, the reliability of collected data will inevitably fluctuate with missed, partial and faulty measurements being unavoidable, resulting in serious potential trouble later on in the workflow such as in the analytics stage. Hence, correct data management (i.e. data cleaning, filtering, transforming and other [10]) actions are mandatory at the beginning of the process. The presence of incorrect data (e.g. incorrectly biased data) is considered a critical challenge because it could potentially lead to incorrect outputs of the analytics engines.
- *Scalability* - considered to be a crucial bottleneck of Big Data solutions. As time goes by, the volume of data being generated is dramatically increasing. Therefore, a key feature that every efficient Big Data processing algorithm must contain is the ability to work with chunks of data that are getting larger every day. Literature [7] states that only incremental algorithms are resistant to this kind of data expansion. Following the problem with processing, storage management is another unavoidable barrier regarding Big Data. Storing the huge quantity of data between its acquisition, processing and analysis requires gigantic memory capacity, thus rendering traditional solutions obsolete. Contemporary cloud-based solutions are also considered to be on the edge of feasibility since responsiveness can be a critical issue, especially in real-time applications, where upload speeds are considered the main bottleneck.
- *Timeliness* - When discussing real-time applications (e.g. stock market, financial fraud detection and transactions parsing, traffic management, energy optimization etc.), quick responses are required and expected practically immediately following data acquisition. Concretely, the supplied information can be rendered completely useless if it is derived with high latency with respect to the collected data. In a money fraud example, the information about an ongoing-scam is only considered important at that specific instance in time in order to potentially stop the questionable transaction before allowing it to proceed.
- *Fault tolerance* - As was previously mentioned, the correctness of the data is considered to be a key aspect of Big Data processing. However, unlike the very frequently used SQL solutions, high volumes, unstructured form, distributed nature of data in NoSQL data management systems and the necessity of near-to-real-time responses often leads to corrupt results with no method being able to guarantee complete validity of the given results.
- *Data security* - especially important when considering Big Data applications on personal or sensitive enterprise data. Depending on the type of data that is being processed, security can sometimes be a crucial component that requires special attention. When considering, for example, a weather forecast or public transport management use case, if a data loss or theft occurs, it can be considered practically irrelevant when compared to a situation where personal information, names, addresses, location history, social security information or credit card PIN codes are stolen because in the latter case data protection must be upheld at the highest possible standard.
- *Visualization* - although the various ways in which the data can be displayed do not affect the data processing segment in any way, visualization is stated in literature as a crucial factor because without adequate representation of the results, the derived knowledge is useless.

In order to successfully resolve the aforementioned challenges, a diverse landscape of solutions has been developed and they will be presented and discussed in the remainder of this paper. However, each of the discussed solutions is tasked with some specific part of the Big Data workflow that can be disaggregated into four main steps *storing*, *processing*, *analytics* and *visualization*. Therefore, in Table I the specified challenges and the appropriate resolutions are presented and the appropriate segment of the Big Data workflow for which that specific issue is attributed is denoted with a plus sign. Given the previously described problems with heterogeneity and data security it is undeniable that storing and processing stages are key when discussing these challenges. On the other hand, scalability and timeliness also influence the analytics stage, because of 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.

Table I – Brief overview of the Big Data workflow

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

IV. BIG DATA FRAMEWORKS

One of the key elements in the all Big data solutions are the underlying frameworks. The first major breakthrough was the **Hadoop** system that is currently supported by Apache. **Hadoop** is highly scalable, robust, reliable, fault-tolerant and allowed for parallel processing on legacy hardware making it cheap to run. The underlying technologies behind **Hadoop** are **HDFS,** high-latency master-slave batch processing file system and the prominent **MapReduce** processing method allowing for performance increases trough data splitting. **YARN**, a popular cluster manager and **Hadoop's** database system **HBase** are also often mentioned in association with **Hadoop.** On the other hand, in addition to batch, faster low-latency stream processing is offered by **Spark** and its libraries like **SparkStreaming**. Big data frameworks are also proposed

trough **Storm** whose structure is often referred to as an acyclic graph with nodes being either spouts (representing data sources) or bolts (representing operations that need to be performed). Another solution in the fast, streaming, real-time ready domain, called **Samza**, proposes a three-layered structure (data streaming with **Kafka**, resource management with **YARN** and processing) whilst **Flink** allows for both slower but more trustworthy batch and faster, more error prone, stream processing.

A. Big Data Storage

Big data solutions also incorporate what is called a NoSQL structure for data management meaning that the underlying processes do not solely rely on SQL and the relational database management systems (RBDMS) but also supplement it with other technologies. Also called non-relational databases, these solutions allow for efficient horizontal (outward) scaling as opposed to traditional vertical (inward) scaling that is required by common SQL operations like joining tables which work extremely slowly in distributed systems. Therefore, NoSQL Big data platforms can easily and cheaply be extended with commodity hardware for new capacities whenever such requirements arise. Non-relational databases can be categorized [11] according to their data storage type into: key-value (**Hazelcast, Redis, Membase/Couchbase, Riak, Voldemort, Infinispan**), wide-colum (**Apache HBase, Hypertable, Apache Cassandra**), document oriented (**MongoDB, Apache CouchDB, Terrastore, RavenDB**) and graph-oriented (**Neo4J, InfiniteGraph, InfoGrid, HypergraphDB, AllegroGrap, BigData**) with literature [12] offering detailed reviews and hands-on experiences with screenshots.

B. Big Data Analytic Tools

Contemporary data driven enterprises do not just stop at data storage but also requires complex processing and information extraction, usually by means of data linking. Knowledge graphs offer huge benefits in this regard by representing data in form of subject, predicate and object data triples, also known as resource description framework (RDF) where subjects and objects are placed in nodes of a graph and the connecting edges represent the predicates. Moving away from RBDMS solutions allowed for flexible schemas with an arbitrary number of nodes without redundant copies and relations. Knowledge graph-based solutions are already employed by Google, Facebook, Microsoft, LinkedIn and Amazon for easier and more efficient data processing and are slowly making their way into other sectors beyond IT like sales and finances.

Big data analytics can be explained as the crucial component of the Big data paradigm as it corresponds to the knowledge extraction from enormous amount of data. Namely, even though processing and storing are challenging enough when these huge chunks of data are supposed to be handled, they exist only to serve the knowledge extraction. Therefore, in this section, the Big data tools and various algorithms for the analytics are going to be discussed and analyzed.

a) Supervised learning

The first group corresponds to the so-called supervised learning approaches – the methodologies which optimize the model's parameter so as to minimize some criterion function which describes the differences between the desired and estimated output. As stated [13], [14], most frequently used supervised techniques are **linear regression** (estimates continuous output with the presupposed linear model with respect to the predictors), **logistic regression** (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** (non-parametric statistical approach used for the regression problems), **discriminant analysis** and **naive Bayes** (also statistical methods, yet used for classification) **neural networks** (able to extract highly informative features even from extraordinary complex problems), **ensemble methods** (improve performances by using combination of various of models), **decision trees** (preferable for classification

Table II - Systematization of regression and classification learning algorithms in Big data tools

	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
Apache Spark	+	+	+	+		+	+	+	+	+							+			
H2O				+					+	+	+			+	+	+	+			
R		+	+	+	+			+	+	+		+	+				+	+		
MOA				+				+									+		+	
Scikit - Learn	+	+	+	+	+		+	+	+	+	+			+	+		+	+		+
Bigml	+				+			+	+	+	+						+			
Weka	+	+	+	+					+				+							

purposes; designed in such a way that the nodes represent the attributes, whilst the branches depict their values) etc. On the other hand, the second group with 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** for the similar reasons as already mentioned etc.

Finally, in this paragraph, apart from the previously presented approaches, commonly used tools will be presented [15]. Expectedly, elementary algorithms are covered by most od frameworks - **Apache Spark**, **Hadoop**, **Apache Mahout**, **H2O**, **R**, **MOA**, **VowpalWabbit**, **TensorFlow**, **BigML** and **Weka. Apache Spark MLLIB** contains logistic regression, decision tree classifier, random forest, gradient-boosted tree, multilayer perceptrons, linear SVM, one-vs-all and naive Bayes. Furthermore, it implements generalized linear regression, decision tree, random forest and gradient-boosted tree for regression purposes, survival and isotonic regression as well. **H20** supports deep learning neural networks, distributed random forest, isolation forest, generalized linear model, gradient boosting machine, naive Bayes, stacked ensembles and XGBoost, whilst **R** [16] covers 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**, Bayesian classifiers, decision trees, perceptrons and drift classifiers which are proposed for data's changing nature over time are available, while **BigML** supports decision trees, random forest, boosting ensembles, logistic regression, deep nets. **Weka** offers linear and logistic regression, naive Bayes, decision trees, kNN, SVM, neural networks, random forest, C4.5. This comparison is presented in Table II, giving a systematic review.

b) Unsupervised learning

Another approach in unsupervised learning is clustering. Having it in mind, here the corresponding frameworks are going to be presented. **Apache Spark**, **Rhadoop**, **Apache Mahout**, **H2O**, **R**, **MOA**, **TensorFlow**, **BigML** and **Giraph** are among suitable choices. Namely, **Apache Spark** implements K-means in the original form, and its bisecting and streaming improvements, Gaussian mixture, hierarchical optimization algorithms, power iteration clustering and latent Dirichlet. **R** offers K-means, Partitioning Around Medoids, clustering large applications, Fuzzy clustering, Model-based clustering and hybrid approaches, whilst **BigML** provides K-means and G-means. **Giraph** contains implemented affinity propagation and K-mean algorithms. Presented clustering tools are shown in Table III.

C. Big Data Visualization

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. The whole workflow of Big Data could potentially lose all purpose if the end result is not represented in an appropriate manner which is why this part is often described as turning data mining into gold mining. Visualization is tasked with an important problem of exploration and explanation of large datasets and thus extracting intelligence and representing it in the manner in which humans generally receive information. Some authors even go as far to state that, visualization can be considered as important as the analytic process because of its close relation to the user. Related literature [17] 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

Table III - Systematization of clustering learning algorithms in Big data tools

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

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.

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 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.

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, 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 are desktop applications for Microsoft 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.

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 [18] and [19] offer concrete analysis on the issue of Big Data visualization.

ACKNOWLEDGMENT

The research presented in this paper is partly financed by the European Union (H2020 LAMBDA project, Pr. No: 809965), and partly by the Ministry of Science and Technological Development of Republic of Serbia (SOFIA project, Pr. No.: TR-32010).

REFERENCES

[1] L. Doug, "3D Data Management: Controlling Data Volume, Velocity and Variety," *Application Delivery Strategies,* 2011.

[2] G. A. Lakshen, S. Vraneš and V. Janev, "Big data and quality: A literature review," in *2016 24th Telecommunications Forum (TELFOR),* Belgrade, Serbia, 2016.

[3] I. Kalbandi and J. Anuradha, "A Brief Introduction on Big Data 5Vs Characteristics and Hadoop Technology," *Procedia Computer Science,* 2015.

[4] M. A.-u.-d. Khan, M. F. Uddin and N. Gupta, "Seven V's of Big Data understanding Big Data to extract value," in *Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education*, Bridgeport, CT, USA, 2014.

[5] N. Khan, M. Alsaqer, H. Shah, G. Badsha, A. A. Abbasi and S. Salehian, "The 10 Vs, Issues and Challenges of Big Data," in *ICBDE,* 2018.

[6] A. Panimalar, V. Shree and V. Kathrine, "The 17 V's Of Big Data," *International Research Journal of Engineering and Technology (IRJET),* 2017.

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

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

[9] A. Oussous, F. Z. Banjelloun, A. A. Lahcen and S. Belfkih, "Big Data technologies: A survey," *Journal of King Saud University - Computer and Information Sciences,* pp. 431-448, 2018.

[10] K. Wadhwani and Y. Wang, "Big Data Challenges and Solutions," 2017.

[11] A. Corbellini, C. Mateos, A. Zunino, D. Godoy and S. Schiaffino, "Persisting big-data: The NoSQL landscape," *Information Systems,* pp. 1-23, 2017.

[12] R. Rai and P. Chettri, "Chapter Six - NoSQL Hands On," *Advances in Computers,* pp. 157-277, 2018.

[13] A. Dey, "Machine Learning Algorithms: A Review," *International Journal of Computer Science and Information Technologies,* pp. 1174-1179, 2016.

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

[15] M. K. Saggi and S. Jain, "A survey towards an integration of big data analytics to big insights for value-creation," Information Processing and Management, pp. 758-790, 2018.

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

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

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

[19] A. Genender-Feltheimer, "Visualizing High Dimensional and Big Data," *Procedia Computer Science,* pp. 112-121, 2018