This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3039455, IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

An Iterative Methodology for Defining Big Data Analytics Architectures

ROBERTO TARDÍO[1], ALEJANDRO MATÉ[2], AND JUAN TRUJILLO.[3],
[1]StrateBI Business Solutions Ltd, Madrid, 28020 Spain (e-mail: roberto.tardio@stratebi.com)
[2]Dept. of Software and Computing Systems, Lucentia Lab, University of Alicante, Alicante, 03690 Spain (e-mail: amate@dlsi.ua.es)
[3]Dept. of Software and Computing Systems, Lucentia Lab, University of Alicante, Alicante, 03690 Spain (e-mail: jtrujillo@dlsi.ua.es)

Corresponding author: Roberto Tardío (e-mail: roberto.tardio@stratebi.com).

ABSTRACT Thanks to the advances achieved in the last decade, the lack of adequate technologies to deal with Big Data characteristics such as Data Volume is no longer an issue. Instead, recent studies highlight that one of the main Big Data issues is the lack of expertise to select adequate technologies and build the correct Big Data architecture for the problem at hand. In order to tackle this problem, we present our methodology for the generation of Big Data pipelines based on several requirements derived from Big Data features that are critical for the selection of the most appropriate tools and techniques. Thus, thanks to our approach we reduce the required know-how to select and build Big Data architectures by providing a step-by-step methodology that leads Big Data architects into creating their Big Data Pipelines for the case at hand. Our methodology has been tested in two use cases.

INDEX TERMS Big data pipelines, Business intelligence, Data Management, Hadoop, NoSQL.

I. INTRODUCTION

IN recent years, the large number of publications on Big Data techniques [1]–[4], technologies [5]–[7] and applications [8]–[10], highlights the importance of the Big Data phenomenon in the field of data processing and analysis. Most researchers [2], [8], [11] define the concept of Big Data using the following features directly or indirectly, also known as the 3 V's of Big Data:

- **Volume:** Huge volumes of data are available for processing, with orders of magnitude of terabytes, petabytes or even exabytes.
- **Velocity:** Strong increase of data generated in (near) real-time and constraints on the time available for processing them.
- **Variety:** Strong increase in the number of data sources which must be integrated with differing and less structured data models.

In addition to the above features, some authors add other V's to the definition of Big data, highlighting Veracity [9], [10], [12] and Value [12]–[14]:

- **Veracity:** Can you trust in selected data sources and providers? Have these data enough quality? Ensuring data quality is a task that sometimes exceeds human capacities.
- **Value:** Complexity to determine a priori the value of the data for the improvement of targeted business processes.

Browsing the literature, we can find a few definitions that include more features, such as Variability [10], [25]. However, we consider these additional features can be deemed less relevant to describe a Big Data scenario since they are included in one of the 5 V's (Volume, Variety, Velocity, Veracity, Value).

By managing and processing data in scenarios where the above features are involved, we are able to implement high impact analytical applications for society [8], [37] and companies [11], [36]. Some of these applications target the improvement of processes in Healthcare, Smart Cities, Smart Vehicles or E-commerce sectors.

Even though the 5 V's have been the main challenge of Big Data for years [6], today the existing technology and methods [5] allow the processing of these data sets in an efficient and effective way. However, several studies [17], [18] identify the lack of professional know-how in the use of Big Data technology and its effective combination to build the system architecture as the main current issues that prevent companies from successfully adopting Big Data. These same studies point as the underlying causes the complexity of Big Data technology and the number of alternative technologies available to implement the same type of processes.

In recent years, several methodological approaches have been published [1]–[4], [15] with the aim to provide effective solutions to the above-mentioned problems. Most of these proposals are based on the analysis of the requirements derived from the 5V's or characteristics of Big Data, which we consider essential to describe a Big Data scenario and a key factor for the choice of the most appropriate techniques and tools for the Big Data Pipeline. However, we observe that existing methodologies either do not guide the development of the Big Data architecture with the sufficient level of detail or they are not applicable to every sector or analytical application.

Based on these findings, in this paper we present an iterative methodology to help IT professionals with the definition and validation of Big Data architectures for analytical applications. This methodology is based on our previous research [16], where we proposed a first version of the methodology and evaluated its application through an Internet of Things (IoT) case study with application in Smart Cities (data analysis of distributed Smart Meters). In this new version of the methodology, we have focused our research on the generation of the Big Data architecture from the requirements of the target Big Data application and we have updated its application to the current technologies and challenges of Big Data, previously commented.

Contributions: We describe a methodology to help IT professionals with the definition of Big Data architectures, that is based on the requirements analysis derived from Big Data features or 5Vs:

- An iterative methodology for the design of Big Data Pipelines, i.e. architectures aimed to support data analytics applications (Section III).
- We identify the main non-functional requirements of a Big Data application. (Section III-C).
- We study the previous requirements for the most current Big Data use cases and sectors (Section III-C).
- An algorithm for the generation of the Big Data Pipeline using a graph visual representation (Section III-C).
- A method for the evaluation of the generated Big Data Pipeline (Section IV).
- Evaluation of the methodology on 2 case studies: a real-world case study of an e-commerce company and also for the IoT case study we presented in the first version of the methodology [16] (Sections III and IV).

In the remainder of this paper, we first review the related work in Section II. Next, in Section III, we will present our proposed methodology for the development of Big Data Pipelines, detailing the application of each phase of it to the e-commerce case study. After this, in Section IV we will present a guide to evaluate the Big Data Pipeline generated and use it to evaluate both e-commerce and IoT use cases. Finally, in Section V, we summarize conclusions and sketch future work.

II. RELATED WORK

For the design of the proposed methodology, we first review the existing methodological approaches for the processing and definition of Big Data architectures [1]–[3], [15], [16], [25]–[27]. Then, we analyze relevant real-world Big Data use cases [8]–[11], [13], [14], [19]–[24], [36], [37] that can benefit from the application of such methodological approaches. Moreover, during the analysis of these applications, we have identified the main features of Big Data that pose challenges to Big Data architectures.

A. METHODOLOGIES FOR BIG DATA

From the academic point of view, there are several methodological approaches that have tried to provide fundamental concepts and theoretical proposals to deal with the challenges posed by Big Data applications such as those reviewed in the previous section. The ultimate aim of these methodologies is to enable and guide the development of these Big Data analytical applications.

First, we reviewed some approaches based on requirements modeling techniques [1], [2], [15]. In [2] a bibliographic review of this kind of approaches is conducted. They conclude that the poor taking of requirements in Big Data projects as one of the causes of failure. They show the utility of requirements modeling techniques to address this issue. However, in spite of its useful conclusions that can be applied to the design of Big Data architectures, they do not propose any methodology. In [1] the authors propose a new framework for the development of Big Data Analytics applications based on business goals (Goal Oriented Requirement Engineering, GORE). This framework consists of a conceptual model to connect business aspects with Big Data processes, an evaluation method and a guide for its application using Spark technology. Although this approach considers the business goals and, therefore, the Big Data requirements of the applications, it does not contemplate the features of the data sources that are also critical to choose the most appropriate Big Data technologies. Similarly, [15] proposes a requirements model for Big Data processing and applies it to a case study. The proposed model considers functional requirements (information requirements) and, like our research, non-functional requirements related to the characteristics of Big Data (V's) data sources. However, despite the advantages of its application, the proposal does not include a method for the selection of Big Data technology and the design of the Big Data Pipeline.

In addition to the above approaches, there are other ones that propose to use requirements modeling in combination with Data Warehousing (DW) techniques [27]. DW techniques were traditionally used for data management, storage and processing in Business Intelligence (BI) systems [28], [29]. In [12], the applicability of existing DW methodologies in Big Data scenarios is analyzed and compared. Although they do not provide a methodological proposal, they highlight the importance of analyzing the requirements of a Big Data application in relation to the features of the data

sources (Velocity, Variety, Volume and Veracity). In [26], a methodology is proposed for the processing of Big Data by applying DW techniques and using Hive (Hadoop tool). It is based on the transformation of multidimensional models into physical data models optimized for Hive. In spite of its advantages, this methodology only takes into account one type of Big Data Analytics application, Data Warehousing, and one technology, Apache Hive. Therefore, unlike our proposal, it cannot be generalized for different case studies and technologies.

Other methodological approaches [3], [4], [25], [32] are based on the analysis of requirements without requirements modeling techniques. Nadal et al. [25] propose a Software Reference Architecture (SRA) than can be used to design Big Data Architectures. As in our research, they identify the requirements of Big data systems in relation to the Big Data main features (Volume, Velocity, Variety, Variability and Veracity). Despite its advantages, this approach does not provide easily measurable metrics and therefore makes it difficult to objectively evaluate the requirements for technology selection. Nor does it consider the nature of the different types of processes that are executed in a Big Data architecture, such as data acquisition, transformation or enrichment ones. Another methodological proposal is the TOREADOR project [32] (Horizon 2020, European project), which presents a methodology that allows users to define the requirements of their target Big Data system through declarative language. From this definition TOREADOR allows to generate automatically the implementation of the architecture using an ontology for the selection of the technologies. In spite of its advantages, the functional and non-functional requirements that are taken into account for the selection of the tools of the pipeline are not detailed until now, focusing more on the automation of the deployment of the generated architecture.

In [3] the author presents a methodology based on MDE (Model Driven Engineering) for the management of a Big Data pipeline with support for the automation of Big Data Analytics processes. To this end, the proposed methodology takes into account the analysis of functional and non-functional requirements (V's of Big Data) and makes use of ontologies to store the configuration of the Big Data pipeline and to automate the execution of processes on it. In addition, it is applied to a case study of cybersecurity. Although this methodology allows us to manage and automate Big Data architectures, it does not provide the necessary steps to carry out the design of it. Despite their disadvantages, these approaches [3], [4], [25], [32] are all very interesting and could be complemented with the advantages of our proposed methodology.

Like the aforementioned proposals, our proposed methodology shares a common starting point based on requirements analysis of Big Data sources and its applications. However, we observe that the existing methodologies either do not guide the development of the Big Data architecture with enough detail or are not applicable to any sector or analytical application. Guiding professionals with enough detail during the design process of the Big Data Pipeline and being a general purpose approach for analytical applications are the main goals we pursue with our proposed methodology.

B. CURRENT BIG DATA APPLICATIONS

From a practical point of view, we have analyzed several real-world projects to identify the main Big Data challenges faced by systems architects and developers. For our review, we have considered the most current Big Data applications [8], [11], [36], [37] in sectors with great impact on society and organizations such as Healthcare, Smart Cities, Smart Vehicles or E-Commerce.

In [9], [13], [19] the features and uses of Big Data in Healthcare are examined. All these studies highlight the large variety of data sources that are available in this sector, where together with structured data (e.g. claims and billing) there is a large presence of semi-structured and unstructured data sources: data generated by machines (vital signs, sensors), biometrics (x-rays and other medical imaging), Electronic Health Records (EHR), clinical test results, admission records, emails, epidemiological data (surveys, statistics) or scientific publications. This Variety of data sources generated at a constant Speed leads to vast Volumes of data, easily reaching petabyte sizes. This is the case of the California healthcare network [13], Kaiser Permanente, with a historical of 30 petabytes over its more than 10 million members. In addition, all these researches highlight that the speed of processing and quality of the information generated are critical. They both can condition the health and even the life of the patient, so that in many cases a real-time processing joined with an interactive query latency are required.

Another field of application of Big Data is the Smart Cities, where we need to implement intelligent services for tasks such as energy management, critical infrastructure [8], [10], [20] or disaster management [22], [23]. Regarding energy management, Smart Grids are a key service in any Smart City, as they provide electricity to the remaining services. The volume of data generated by a Smart Grid is very high [20], estimating that a Smart Grid with 1 million users with Smart Meters can generate approximately 1 Tb of data per day. Those terabytes of data stored in tabular format result in hundreds of millions of rows [10]. In addition to Smart Meters, we have a wide Variety of data sources: Sensors, Substations, Meters, Supervisory Control And Data Acquisition (SCADA) systems or historical data. Most of these data sources are semi-structured or structured in nature [20] and contain data as diverse as voltage data, control devices, events (e.g. breakdowns, voltage loss), network equipment data or historical data. In addition, Smart Grids often require real-time processing, as some data has to be moved and processed in a critical time in order to preserve its Value. This is the case of data related to electricity production (e.g. photovoltaic) and demand, where [10] technologies such as Apache Kafka or Storm are being used for real-time acquisition and processing. Veracity will depend on the application. For example,

if we lose some samples or generate some duplicates of energy consumption and production data it will not have a great impact, due to the large number of samples collected. However, when trying to identify breakdowns or other critical situations in the network, the loss or duplicity of data can be less tolerated.

Other Big Data applications for IoT are those related to autonomous communication between vehicles [36], [37] or Internet of Vehicles (IoV). These applications allow new opportunities for vehicle manufacturers and service providers, such as route planners, collision warning systems, traffic monitoring or passenger infotainment. In [36] the authors propose to optimize communications vehicle-to-vehicle (V2V) by calculating the probability of communication between nearby vehicles from data provided by communication systems and the social relationship tightness that represents content selection similarities on real-world social big data. The proposed Big Data architecture combines batch processing of up to terabytes of data from social networks, with real-time processing of the data provided by the vehicle. In [37] the authors present another approach to the same question but propose the use of a coalition formation game model on vehicle-provided speed and location data, in combination with a GIS (Geographic Information System). However, in both cases the validation only focuses on the results of the proposed methods and not on the evaluation of the requirements of the Big Data architecture. Therefore, we recommend for both cases [36], [37] the application of our proposed methodology in order to validate and, therefore, successfully implement these architectures.

Finally, we analyze e-commerce [11], [14], [21], one of the sectors where the Big Data application is more consolidated. Some of the Big Data applications in e-commerce are decision-making process improvement, market segmentation and the identification of needs and innovations in the product/market/business model [11]. To implement these Big Data applications, e-commerce companies make use of the great Variety of data collected from its customers: clicks and impressions in advertising elements, transactions, product ratings, location or social networks. Most of these sources [21] are structured (e.g. charges) or semi-structured (e.g. shopping cart), but companies such as Amazon, eBay or Expedia are already using other sources less structured from social networks such as photos, notes, blog entries, web links and news. This Variety together with the typical international expansion of these companies, leads to huge data volumes in the order of tens of terabytes in many cases and even petabytes [14]. For example, e-Bay, the world's largest Business-to-Consumer (B2C) platform with hundreds of millions of active sellers, buyers and products, stores data in Hadoop environments sized to manage more than 100 PB. As for Velocity, [21] emphasizes that efficiency in the storage and query of data is critical. For this reason, e-Bay uses the Apache Kylin tool to execute queries on Terabytes of data stored in Hadoop with response times under the second. Another example of Velocity is Amazon [21], where they have developed dynamic pricing systems that adjust prices every 15 seconds using data from competing e-commerce sites. Finally, the Veracity of the data is very important in e-commerce. However, there are processes in which a certain margin of error is tolerable, such as those related to click-stream analysis, clicks and impressions.

Although the use cases reviewed vary in their functional requirements (e.g. targeted insights), the set of non-functional requirements to be considered when implementing their Big Data architectures is quite similar for all of them. As we can observe, these non-functional requirements are directly related to the 5V's of Big Data, thus we can use the result of their analysis to select the most appropriate Big Data technologies and techniques for each use case. This way, we can ensure that the Big Data architecture implemented will be able to cope with the non functional requirements of the targeted analytical applications, thus facilitating its successful implementation.

III. PROPOSED METHODOLOGY

In this section, we present the proposed iterative methodology for the definition of Big Data Pipelines (architectures) to support Big Data analytical applications. Our methodology is composed of 5 phases, each one composed of multiple sub-processes.

First, during phase 1, non-functional requirements derived from the 5V's of Big Data are identified. These requirements are used as an input for our proposed algorithm to generate an initial structure of the Big Data Pipeline. Afterwards, this structure is successively refined during the remaining phases (2-5), where the selection of the most appropriate technologies and techniques is performed based on the analysis of the requirements identified during phase 1. Moreover, as this methodology is envisaged as an iterative methodology, in Section IV we describe the evaluation process that can be performed to refine the design of the Big Data Pipeline by iterating again through the different phases of the methodology.

Fig. 1 shows a graphical representation that summarizes the methodology, composed of 5 phases, sub-processes and features of Big Data that guide the execution of each of the phases. In the following, we describe each phase of the methodology using a running case study based on an e-commerce company.

A. REAL USE CASE: E-COMMERCE

The methodology proposed is an evolution of the one presented in our previous research [16] that was initially applied in a case study of IoT-Smart Cities, for the analysis of distributed Smart Meters data. Since the methodology has evolved, and our purpose is to make it suitable for any Big Data Analytics scenario, the new version of our methodology has been evaluated with a new case in a real e-commerce case study as well as with our previous case study.

The e-commerce company under study develops and manages pay-per-subscription services in more than 15 countries

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3039455, IEEE Access

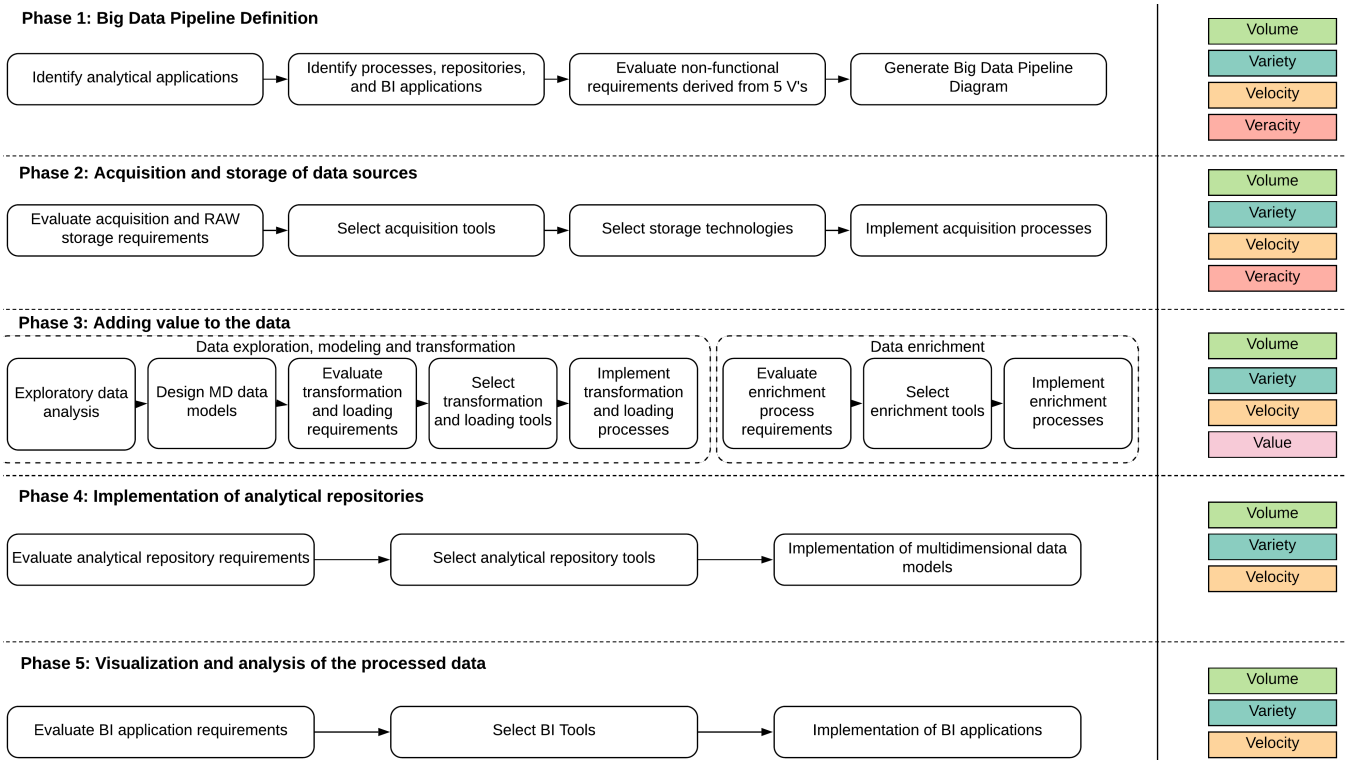

FIGURE 1. Proposed methodology for the definition of Big Data Pipelines.

and its marketing campaigns generate huge volumes of data at a constant rate. As a result of this digital activity, a large variety of data sources are available:

- **DS1:** Customers, subscriptions and unregistrations database with 4 tables. Each of these tables stores about 40 million rows of historical data. In addition, the data persistence is centralized, allowing to implement the acquisition of these data in real-time, if necessary, using lightweight and semi-structured formats such as JSON.
- **DS2:** Billing database stores payments in a table of about 2,200 million rows. Each month 20 million new rows are incrementally added to this table.
- **DS3:** Database of customer acquisition web elements (landing pages, banners or advertising elements) and content pages accessed from the above. These data about the services and marketing are stored in 30 tables with a volume of about 100,000 rows.
- **DS4:** Web servers log containing records of impressions and clicks on web pages (landing pages, banners or advertising elements) along with other information gathered from the visitor (provider, payment method, type of device). Currently, this data is persisted in flat log files and processed daily. It is in semi-structured format and after being processed generates about 4 million rows per day, which translates into about 1,460 million rows per year. The current history is about 7.3 billion rows. The processes to tackle these log files are becoming a bottleneck in the current company architecture.

The target analytics applications are:

- **A1:** Dashboard and reporting for interactive analysis of metrics related to active customers, subscriptions, unsubscriptions, charges, impressions, clicks and associated metrics, such as click rate (CTR) or conversion rate (CR), filtering by services and their most relevant context data, such as country. In the current architecture, the ETL processes for updating data are executed in a non-distributed way. They take between 30 minutes and 2 or 3 hours to complete execution. Similarly, the most complex analytical queries latency is greater than 20 seconds. The goal for the new version of the architecture is that the data can be refreshed with a frequency less than 15 minutes and query latency should be less than 5 seconds and never take more than 10 seconds.
- **A2:** Forecasting process for subscriptions and unsubscriptions. The data must be generated daily by the trained statistical or machine learning models and must be stored to enrich the reporting/dashboarding interactive (A1) repository.
- **A3:** Dashboard for the real-time analysis of active customers, subscriptions and un-subscriptions, with up-to-date data that is refreshed within less than 60 seconds. Speed is prioritized over data quality, allowing an error of up to 3% deviation from real values.

It is also noteworthy that the e-commerce company had an IT team with high knowledge of relational databases, Data Warehousing (DW) and reporting. However, their team had

hardly knowledge of other Big Data technologies such as the Hadoop environment.

B. TECHNOLOGY

One of the main goals of our proposed methodology is to guide the selection of the most suitable technology for our big data scenario. We can classify the current and most deployed Big Data technologies [5] into two main groups:

- **Hadoop framework tools:** A set of open source tools widely used for the storage and distributed processing of Big Data. Some of the most known Hadoop tools are Hadoop Distributed File System (HDFS), Map Reduce, Spark or Kafka.
- **Other NoSQL tools:** Tools for data storage and processing that allow high flexibility in data schemes and query languages to support Big Data scenarios. Hadoop tools can be considered NoSQL, but there are other NoSQL tools that are also widely used in Big Data architectures such as Mongo DB or Apache Cassandra.

From a technological point of view, Hadoop is an environment that has all the necessary tools to carry out Big Data tasks. For this reason, we recommend pre-selecting Hadoop as the default environment. However, there are other NoSQL technologies (e.g. Apache Cassandra, Mongo DB) that perform more specific tasks more efficiently or with greater capacities. Therefore, our methodology takes into account these capabilities in order to select the most appropriate tool for each process in each of the phases, whether from the Hadoop environment or not.

Furthermore, to exemplify our methodology we have chosen mainly open source technologies, which are widely used and easily accessible for evaluation. However, we also recommend considering existing enterprise technologies to apply our proposed methodology. This recommendation includes both on-premise technologies and cloud services. For example, Microsoft Azure or Amazon Web Services (AWS) provides a wide range of cloud services to cover any phase of our proposed methodology and are often compatible with Hadoop technologies.

C. PHASE 1. BIG DATA PIPELINE DEFINITION

In a Big Data project, it is common to define one or more target analytical applications, such as a reporting system that supports analysis of key metrics for the financial area, or a dashboard for the analysis of impressions and clicks in the marketing area. Each of these applications requires one or more data sources that, in most cases, need to be preprocessed.

Considering the above, a Big Data Analytics application is defined by its application requirements, but at the same time by the non-functional requirements of its data sources. Moreover, it is possible to establish a relationship between those requirements and the 5Vs of Big Data (Volume, Variety, Velocity, Veracity and Value). These requirements determine the complexity and form of the processes that need to be implemented and also the desired features of the tools needed to implement these processes. Therefore, our methodology proposes to build the Big Data Pipeline based on the analysis of non-functional requirements derived from the 5Vs of Big Data.

With this aim, in Table 1 we have identified the main (non-functional) requirements and its common values for categorization from the analysis of the current Big Data applications reviewed in Section II [8]–[11], [13], [14], [19]–[24], [36], [37]. These requirements are the basis of our methodology, as they will determine the Big Data Pipeline design.

Non-functional requirements identified in Table 1 are those that are directly related to the V's of Big Data and also critical for the definition of the Big Data Pipeline. However, there are a number of non-functional requirements that are not derived from the Big Data V's but that complement the requirements identified in Table 1. We have identified i) the know-how required for the use of the Big Data tools, and ii) the compatibility (integration capabilities) between these tools, as the most critical non functional requirements that cannot be derived from the Big Data V's. However, the in-depth analysis and application of such requirements are beyond the scope of this research, leaving the user the possibility of evaluating them according to the specific nature of the scenario or organization of the application.

It is noteworthy that Veracity and data quality include several characteristics, such as accuracy, availability, or consistency among others. But these features are beyond the scope of the "loss of data and generation of duplicates" requirement (R8) because they require a more in-depth analysis that we would like to address in future versions of the methodology.

Table 2 exemplifies the use of Table 1 to evaluate the requirements of applications in the sectors where Big Data technologies and techniques are changing the way that analytics are approached. These sectors are extracted from the set of Big Data analytics use cases reviewed in Section II to determine the requirements values. This table can serve as a reference for the application of the proposed methodology to use for cases in any of these sectors.

As we have done in Table 2 with current Big Data analytics use cases, the first step to apply the proposed methodology to our targeted case study is to evaluate the requirements proposed in Table 1 for each target application and its data sources. In addition, we must consider that it is very common to store data acquired in raw format [27]. This would allow users to iterate over methodology phases for extracting new information, initially not identified, that can be useful for the target applications. These new insights could serve to improve current analytics applications or create new ones. Storing data in raw format is also useful to enable different processes to access the data and extract it as they need (fields, format) for the specific application they aim to support, without having to replicate the acquisition process.

Taking into account the above guidelines, we conducted the evaluation of the requirements proposed in Table 1 for the applications in our e-commerce case study. In our case study,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3039455, IEEE Access

TABLE 1. Proposed non-functional requirements of a Big Data Analytics application

Big Data Feature	Derived Non-Functional	Common Categorization Values
Volume	R1. Combined size of all data sources	**Low:** <1 Terabyte **Medium (Med):** $1\ Terabyte \leq x \leq 1$ Petabyte **High:** $> 1 Petabyte$
	R2. Combined No of rows of all data sources	**Low:** $< 100\ million\ rows$ **Medium (Med):** $100\ million\ rows \leq x \leq 1$ billion rows **High:** $> 1\ billion\ rows$
Velocity	R3. Processing opportunity	**Real Time (RT):** $< 1 minute$ **Near Real Time (NRT):** $1\ minute \leq x \leq 15\ minutes$ **Batch:** $> 15\ minutes$
	R4. Data querying latency	**Interactive:** <= 5 seconds **Semi Interactive:** $5\ seconds \leq x \leq 15\ seconds$ **Batch:** $> 20\ seconds$
Variety	R5. Number of different data model	**Low:** $< 5\ data\ models$ **Medium (Med):** $5\ data\ models \leq x \leq 20\ data\ models$ **High:** $> 20\ data\ models$
	R6. Structure Complexity (multi valuable)	**Structured (Struct.)** , **Semi-Structured** , **Unstructured (Unstruct.)**
	R7. Complexity of the designed analytical models (phases 3 and 4)	**Low:** 1 Table (de-normalized data) **Medium (Med):** $1 \leq Tables \leq 15$; $1 \leq Joins \leq 15$ **High:** $Tables > 15; Joins > 15$
Veracity	R8. Data quality (Duplicates and data losses)	**Low:** Duplicates and data loss allowed **Medium (Med):** Duplicates or data loss allowed **High:** No duplicates nor data loss

TABLE 2. Evaluation of the requirements identified in Table 1 for current Big Data applications

Sector	Key Data Sources	R1	R2	R3	R4	R5	R6	R7	R8
Healthcare	Elec. Health Records, images and sensors	High (Petabytes)	High	Real Time	Interactive	High	Semi Structured	High	High
IoT-Infrastructures	Smart meters, sensors and SCADA	High (Petabytes)	High	Real Time	Interactive	Medium	Semi Structured	High	Medium
IoT-Disaster Relief	Social media and sensors	Medium (Terabytes)	Medium	Real Time	Interactive	Medium	Semi Structured	High	Medium
IoT-Vehicles (IoV)	Vehicle (network, GPS, sensors, images), social media and GIS	Medium (Terabytes)	High	Real Time	Interactive	High	Semi Structured and Unstruct.	Medium	High
E-Commerce	Transactions, impressions and social media	High (Petabytes)	High	Real Time	Interactive	High	Semi Structured	High	Medium

we have identified A1, A2 and A3 as the applications to be implemented on the data sources DS1, DS2, DS3 and DS4. In addition, we have added an application that represents the repository of raw data, identified as A0. The result is shown in Table 3.

The applications are also composed of different processes, repositories, BI applications and connections between them, which represent the data flows. In Table 4, we identify and describe all the elements that we propose for the generation and representation of the Big Data Pipeline.

Using the notation proposed in Table 4, we propose to represent the Big Data Pipeline as a graph. Moreover, for the creation of the Big Data Pipeline, we propose the algorithm in Fig. 1, which uses as input the requirements analysis of each application from our targeted case study and generates as output the Big Data Pipeline.

Applying the algorithm in Fig. 1 on the analysis for the applications requirements in our e-commerce case study, as shown in Table 3, we generated the graph of the Big Data Pipeline pictured in Fig. 2.

The Big Data Pipeline generated in this first phase defines the processes, repositories and BI applications of our Big Data scenario. In the following phases (2-5) we will evaluate the requirements of each of these elements of the Big Data Pipeline to determine the most appropriate techniques and technologies for implementation. This way, the final result of the application of the proposed methodology will be the

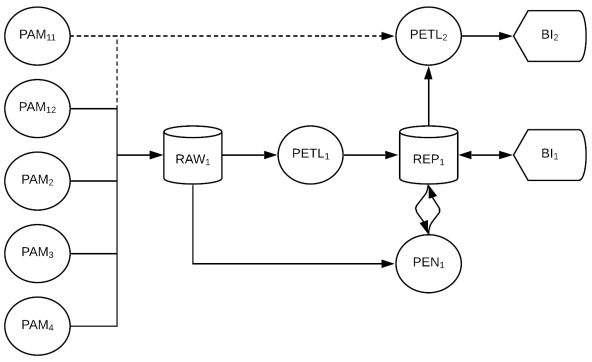

FIGURE 2. Big Data Pipeline for the e-commerce case study after the application of phase 1.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3039455, IEEE Access

TABLE 3. Elicitation of requirements for the e-commerce case study

Analytics Application	Data Sources	R1	R2	R3	R4	R5	R6	R7	R8
A0.Raw storage	DS1, DS2, DS3 and DS4	Med	High	Near Real Time	Batch	High	Structured	Low	Med
A1.Dashboard and reporting	DS1, DS2, DS3 and DS4	Med	High	Near Real Time	Interactive	High	Structured	High	High
A2.Forecasting	DS1, DS2, DS3	Med	Low	Batch	Interactive	Low	Structured	High	High
A3.Real Time Dashboard	DS1	Low	Low	Real Time	Interactive	Low	Semi structured	Low	Med

To calculate the size (R1) we use the average size of each record in the source system, which is about 20 Kb.
Data from the DS1 source is delivered in real-time in a semi-structured format typically used in streaming acquisition and movement systems.

TABLE 4. Proposed notation for the Big Data Pipeline graph representation

Element	Notation	Description
Acquisition and movement process	PAM	Processes for data acquisition from data sources in order to move it to repositories and processes. They also enable the movement of data between processes or repositories.
Transformation process	PETL	Processes used to structure and integrate data from different sources and load them into the analytical repository.
Enrichment process	PEN	Processes data that allow enrichment through the extraction of new insights by using Machine Learning (ML) techniques or by the integration of current data with no initially considered data sources.
Analytical Repository	REP	Storage and support for data queries generated by BI applications or other processes.
Raw Storage	RAW	Storage of data in raw format.
BI application	BI	Support for Business Intelligence (BI) applications such as dashboards, reports and OLAP views to enable analysis of data stored in repositories or acquired from processes.
Batch data flow	→ ↔	Represents a batch data flow between two notations. The direction of the data flow is indicated by the arrow at the end. Bi-directional flow can be represented with arrows at both ends.
Streaming data flow	⇢ ⇠⇢	Represents a streaming data flow between two notations.

Big Data Pipeline with the processes, repositories and BI applications labeled with the technologies selected thanks to the application of the methodology.

Algorithm 1: BIG DATA PIPELINE INITIALIZATION

Input: Table *R* with the analysis of the requirements of each analytical application *A*
Output: *Elements*: A list of *Element* representing processes or repositories;
Pipeline: A list of *Pairs* representing the connections between two different *Elements*

```
For all data source F into table R do
    Elements.add(PAM)
end for
Elements.add(RAW)
For all acquisition process PAM into Elements do
    Pipeline.add(PAM, RAW)
end for
For all application A into table R do
    Elements.add(PETL)
    For all data source required by PETL do
        Pipeline.add(RAW, PETL)
    end for
    if requires-persistence(A) then
        Elements.add(REP)
        Pipeline.add(PETL, REP)
    end if
    if requires-enrichment(A) then
        Elements.add(PEN)
        For all data source F required by PEN do
            Pipeline.add(RAW, PEN)
        end for
        if requires-persistence(PEN) then
            if Elements.notContains(REP) then
                Elements.add(REP)
            end if
            Pipeline.add(PEN, REP)
        end if
    end if
    if requires-BI-application(A) then
        Elements.add(BI)
        For all data source F required by BI do
            Pipeline.add(REP, BI)
        end for
    end if
end for
```

D. PHASE 2. DATA ACQUISITION AND STORAGE

The goal of this second phase is to identify the most appropriate technologies for the implementation of data sources acquisition and movement processes (PAM) and for the raw data repository.

Regarding the acquisition of data sources, first of all, we have to identify the most suitable requirements of the Table 1 to determine the choice of data acquisition technologies. Based on the review of current Big Data applications in Section II, and also in our professional experience in the field, we identified the following requirements as the most relevant:

- **Volume:** It is very important that the chosen data acquisition technology does not become a bottleneck due to the volume of data (R1) received from the data source, to allow the acquisition of data complying with the required processing opportunity (R3).
- **Velocity:** It is necessary that the technology allows the data generated in the data sources to be available in the repository or destination process in the time determined

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3039455, IEEE Access

by the processing opportunity (R3). Although there are tools that support both real-time and batch loading, we will only consider the modes for which a tool is more suitable or has better capabilities.

- **Variety:** No. of data models (R5) and its structure complexity (R6). The available technologies differ in their flexibility regarding the structure of the data they can process. For example, general purpose ETL tools such as Pentaho Data Integration or Talend support data acquisition from a variety of sources, unlike others such as Sqoop that are limited to massive loads of structured data stored in databases.
- **Veracity:** We have to consider the quality constraints (R8) of the application as to whether or not to allow the generation of duplicates or loss data during the acquisition process. These data quality problems are typical of real-time data acquisition processes, where only some tools such as Apache Kafka guarantee that none of the above problems occur (exactly once semantics). There are other data quality characteristics that are outside the scope of the requirement "loss of data and generation of duplicates" (R8). However, we do not include these characteristics into our table because they are maintained through techniques that are not specifically related to the tool of choice, unlike the generation of duplicates and data loss.

In order to continue with the application of the methodology, we have to evaluate the above mentioned requirements for existing Big Data technologies. In our case, we have selected and evaluated some of the most commonly used data acquisition technologies [10], [14], showing the result in Table 5.

After the evaluation of technologies, we have to determine what tools are more suitable for each acquisition and movement process (PAM) in our Big Data Pipeline. To this aim, we propose to follow the subsequent steps:

- Evaluate the identified acquisition requirements (R1, R3, R5, R6, and R8) for the PAM process taking into account the processes, repositories, or BI tools that will use this PAM process as a direct source.
- Then, we can use the technology evaluation we performed in Table 5 to select the tool that meets the requirements determined for each data source. In case there are alternative tools for the same data source, we can evaluate these against the complementary requirements of required know-how and compatibility of tools.
  - **Know-how:** The required knowledge and learning curve to use the tool, considering the skills of our IT team.
  - **Compatibility:** The degree of compatibility of the candidate tool with the rest of tools selected for our architecture during the first iteration of the methodology.

Table 6 shows the result of the application of the previous steps to our e-commerce case study. The IT team from the e-commerce company already had some know-how of Kafka. Therefore, although Flume was an alternative, our methodology selects Kafka as the most adequate technology for Real-Time acquisition of the DS1 source (PAM_{11}) used in the $PETL_2$ process.

TABLE 5. Evaluation of Big Data acquisition technologies

Tool	Volume R1	Velocity R3	Variety R5	R6	Veracity R8
Apache Kafka	High	RT	High	All types	High
Apache Flume	High	RT	High	Semi & Unstruct.	Med
Apache NiFi	High	Batch - RT	High	All types	High
Apache Sqoop	High	Batch - NRT	High	Struct.	High
Pentaho DI	Med	Batch - RT	Med	All types	High
Talend	Med	Batch - RT	Med	All types	High

TABLE 6. Evaluation of the acquisition processes and tool selection

PAM	Volume R1	Velocity R3	Variety R5	R6	Veracity R8	Selected Tool
PAM_{11}	Low	RT	Low	Semi Struct.	High	Kafka
PAM_{12}	Low	NRT	Low	Struct.	Med	Sqoop
PAM_2	Med	NRT	Low	Struct.	High	Sqoop
PAM_3	Low	NRT	Med	Struct.	High	Sqoop
PAM_4	High	NRT	Low	Struct.	Med	Sqoop

In addition to the use of the most suited acquisition tools, we recommend implementing the raw repository in phase 2. In order to carry out this implementation, we analyze the requirements of Table 1 that are most relevant for the selection of the technology for the raw storage in our Big Data Pipeline:

- **Volume:** We must consider the size (R1) regarding the scalability of repository storage technology. For example, HDFS enables distributed and scalable storage from 1 to N storage nodes, thus the scalability of this storage technology is very high. On the other hand, AWS S3 and Azure Blob Storage are cloud-based object repositories, having very high scalability and being easy to manage (automatic scaling without adding physical nodes or disks). Furthermore, they support the integration with all Hadoop based technologies and also with other NoSQL systems.
- **Velocity:** We must evaluate the processing opportunity (R3). For this purpose, we have to consider the read and write rates of the storage systems, as well as the possibility of using the data locality principle. In the case of HDFS, we can decide the disks that will be used as storage. Therefore, the write/read rate will depend on the features of the disks we use (e.g. the use of Solid-State Drives). In addition, HDFS favors complying with the data location principle, thus enabling subprocesses (i.e. MapReduce jobs) that can be moved to the machines where the data is located. This reduces network traffic in a Hadoop cluster and therefore helps to improve processing efficiency. In the case of using object stores technologies such as AWS S3 or Azure

TABLE 7. Evaluation of storage technologies to implement the raw data repository

Tool	Volume R1	Velocity R3	Variety R6	Veracity R8
HDFS	High	Batch - RT	All types	High
S3	High	Batch - NRT	All types	High
Azure Blob Storage	High	Batch - NRT	All types	High
Local file system	Med	Batch - RT	All types	Low

Blob Storage, the principle of data location cannot be guaranteed, thus getting worse read and write rates. With these performance differences in mind, object storages are sometimes used to complement HDFS with lower cost and ease of scaling, storing less frequently accessed data or data sources with more relaxed processing opportunity (R3).

- **Variety:** Since the purpose of raw data repositories is to store data in its original format, the fewer restrictions over the Variety (R6) of supported formats and data structure, the more appropriate the technology will be to implement this type of repository.
- **Veracity:** The quality of the data for the raw storage will be evaluated in terms of tolerance to failures in the storage system in order to avoid data loss (R8). In the case of HDFS and other NoSQL technologies with distributed storage, there is a replication factor number that indicates on how many cluster machines the data will be replicated. The higher the replication, the lower the probability of losing data. If we use local disk storage there is no such data replication unless we implement it, so it is a less failure-tolerant option.

After analyzing the aforementioned requirements to determine the most adequate raw storage technology, we have evaluated some of the most commonly used storage technologies in current Big Data architectures. The result is shown in Table 7.

We have applied the repository evaluation in Table 7 to determine the technology to be used in our e-commerce case study. According to the results, we have opted for the combination of HDFS and S3. HDFS is used to store the most recent data and the intermediate results of the processes (Map Reduce, Tez). Meanwhile, S3 is used to store the less recent and consolidated data. Both HDFS and S3 enable the access to the data from different applications of the Hadoop environment such as Hive, Spark or Kylin.

Once we have selected our tools for data source acquisition and the raw repository, we refine the Big Data Pipeline graph by tagging the PAM processes and raw repositories with the selected technologies. The resulting Big Data Pipeline is shown in the Fig. 3.

E. PHASE 3. ADDING VALUE TO THE DATA

The aim of the third phase is to add value to the data acquired by exploring and processing it. In most cases [26]–[28], data structuring techniques are used to add value to the data. We can structure the data by means of two types of processes:

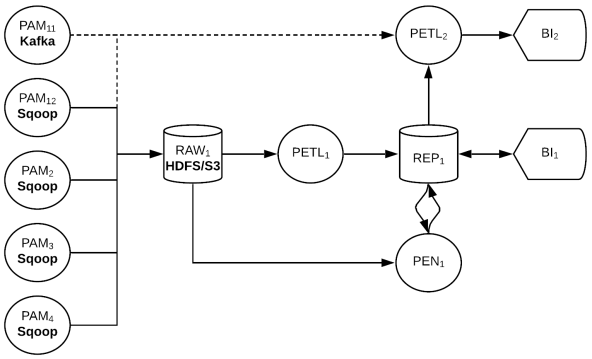

FIGURE 3. Big Data Pipeline for the e-commerce case study after the application of phase 2.

- **Data exploration, modeling and transformation:** We have to explore raw data sources stored in the raw repository or acquired in real-time through streams. Afterwards, we have to integrate them by means of data models. As a result of this process, the most appropriate tools and techniques will be selected to implement the PETL transformation processes together with the most adequate technologies to implement the analytic repositories according to the data models.
- **Data Enrichment:** Data enrichment is achieved through the extraction of new knowledge in the data and through the integration of processed data with data sources that had not been yet considered. The extraction of new knowledge is achieved through analysis and Machine Learning (ML) techniques. As a result of this process, we will select the most adequate tools to implement our PEN enrichment process.

1) Data exploration, modeling and transformation

First, in order to extract useful knowledge from the data sources, we need to explore each source to identify its structure and determine those data that are relevant for our target analytics applications.

Furthermore, some applications require the integration of data obtained from multiple data sources. It is in the integration processes where one of the most relevant features of Big Data is presented: Variety or heterogeneity of the data sources. It is very common to require the integration of data from sources with different data models, file formats and structure levels, through explicit (in the best case) or implicit relationships between them.

To this end, we agree with other authors [26]–[28] that Multidimensional Modeling (MD) techniques are very useful for this process due to their simplicity and efficiency:

- **Simplicity:** Since we can represent the domain of the problem by means of facts of study and dimensions or context of analysis.
- **Efficiency:** They allow to store the processed data in simple schemata like the star schema, which facilitates

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3039455, IEEE Access

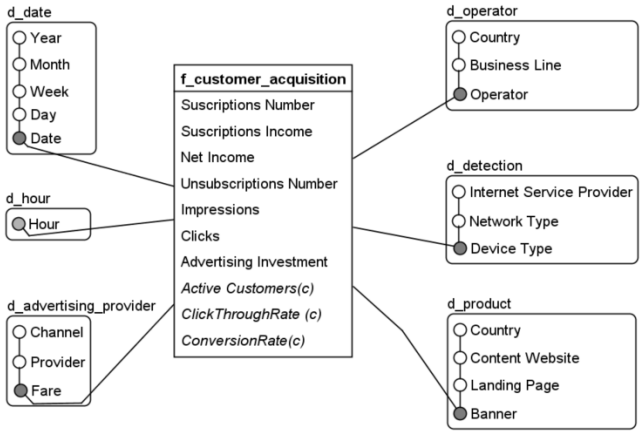

FIGURE 4. Conceptual representation of the multidimensional star model for the e-commerce case study. A few metrics from the f_customer_acquisition fact table are calculated (c) from the remaining direct metrics.

the efficient processing, storage and querying using the repositories and BI tools that we will select in phases 4 and 5.

We consider that the application of DW techniques in Big Data should be carried out in an iterative manner [16], [27]. This enables us to refine the initial data models by adding new valuable information that contributes to achieve application goals or to improve the existing business processes of the organization.

By applying the exploration of raw data sources and DW techniques for the A1 application in our e-commerce case study, we have obtained the multidimensional (MD) model shown in Fig. 4. The process followed is described with more detail in our previous methodology version [16]. In this MD model, we have identified the need to analyze the number of clicks and impressions (fact table named f_customer_acquisition) by advertising element of each service provided (dimension d_product), aggregated by years, months, days (d_date) and hours (d_hour), through a kind of device (d_detection) that, for instance, could be a mobile, tablet or PC.

We can use the same Big Data tool to carry out both the exploration processes and the transformation of data in the designed MD models. Tools that allow data transformation usually also allow data exploration (e.g. Apache Spark, Hive). To this end, we propose to evaluate the candidate tools for data transformation according to the following requirements:

- **Volume:** The size of the data (R1) from all Big Data sources to be processed and combined in the same process can lead to performance and stability issues (e.g. due to RAM usage and processing capacity) in traditional non distributed processing technologies such as Pentaho Data Integration (PDI), Talend or ad-hoc ETL processes (e.g. using Java or PHP programming languages). Distributed processing technologies such as Map Reduce or Spark enable to implement data transformation and loading processes on data sets with Big Data features. In addition, technologies such as PDI[1] and Talend can run on Spark in its latest versions, thus benefiting from the advantages of distributed execution.
- **Velocity:** A good query latency (R4) with Big Data data sources will allow us to be more efficient when discovering the relevant information for the definition of our multidimensional models (MD) by applying DW techniques. Moreover, the chosen tool must allow the implementation of the PETL processes within its processing opportunity constraints (R3). In this sense, we will assess the efficiency of the tool considering both the processing of historical data and the incremental processing of new data. For example, Apache Spark is very efficient for both batch and real-time processing. However, other tools such as Kafka Streams are very efficient for real-time processing, but less suitable for processing historical data.
- **Variety:** We must evaluate the flexibility to read and integrate the different data sources acquired. Data sources can vary in terms of data model, file format, and structure level. Therefore, we should evaluate transformation tools against the variety of data sources they support (R6). Big Data tools from the Hadoop environment such as Map Reduce, Hive, Spark, Flink or Kafka Streams allow efficient processing of data sources with any level of structure.

TABLE 8. Evaluation of data processing technologies to implement PETL processes

Tool	Volume R1	Velocity R3	R4	Variety R6
Pentaho DI	Med	Batch	Batch	Struct. & Semi Struct.
Talend	Med	Batch	Batch	Struct. & Semi Struct.
Map Reduce	High	Batch	Batch	All types
Hive	High	Batch	Semi Interactive	Struct. & Semi Struct.
Spark	High	Batch – RT	Semi Interactive	All types
Kafka Streams	High	RT	Interactive	All types
Flink	High	Batch – RT	Interactive	All types
Storm	High	RT	Interactive	Semi Struct. & Unstruct.

After identifying the aforementioned requirements, in Table 8 we have evaluated some of the most commonly used tools [5] that enable both raw data exploring and implementing the processing required to transform the data and load the resulting data into the defined MD models. Although some of the tools evaluated also have features for data acquisition (e.g. Kafka or Pentaho Data Integration), we have not taken these features into account for this phase 3, but only for phase 2.

[1]https://help.pentaho.com/Documentation/7.1/0P0/Setting_Up_AEL

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3039455, IEEE Access

TABLE 9. Evaluation of the transformation processes and tool selection

Process	Volume R1	Velocity R3	 R4	Variety R6	Selected tool
$PETL_1$	High	NRT	Not Specified	All types	Spark (SQL)
$PETL_2$	Low	RT	Not Specified	Struct. & Semi Struct.	Spark (Streaming)

In addition to the requirements evaluated in Table 8, we recommend to evaluate the required know-how as well as the compatibility with other tools in the Big Data Pipeline. On the other hand, query latency restriction is not usually indicated for data exploration (R4). Therefore, we recommend evaluating the query latency as a tiebreaker criterion in those cases, since it will allow us to improve productivity while performing this task.

After evaluating Big Data technologies for data transformation, we have to evaluate the requirements for each PETL process in our Big Data Pipeline. To this end, we can use Table 8 to select the most suitable tools for each PETL process.

In our e-commerce case study, we have identified two different PETL processes. In Table 9 we have analyzed their requirements and then, according to the results, we choose the best-suited tool for each PETL process.

For process $PETL_1$, we selected the Spark tool using its SQL module. On the other hand, process $PETL_2$ requires processing data in real-time, allowing this stream to be combined and aggregated with the historical data stored in the REP_1 repository. Considering this fact, tools such as Apache Storm or Kafka Streams have been discarded because, despite their real-time features, they are not designed for the integration of stream processing with large volumes of historical data [33]. Moreover, these tools require a greater know-how than Spark SQL. Thus, we have identified Spark (Spark Streaming + SQL) and Flink as the most appropriate tools to implement $PETL_2$.

However, it is recommended to use only one tool –Spark- to implement all the PETL processes in our Big Data Pipeline for maintainability reasons. Therefore, we have chosen Spark as it reduces the overall know-how required.

2) Data enrichment

In addition to the PETL processes for the transformation of data sources, we have to implement the PEN enrichment processes identified so far. We can classify these processes into two subtypes:

- **Addition and integration of new data sources:** For example, in our case study we have IP addresses relative to the impressions and clicks captured in advertising elements (e.g. banners) from the different payment services offered. Obtaining geographic information from the provider of this IP using an open service such as ipdata.co, can give us additional and useful information about visitors or potential customers, such as country, region, city, zip code, internet service provider or proxy detection.
- **Use of Machine Learning techniques:** Another way to add value to the data is extracting hidden knowledge automatically through the application of algorithms and statistical models, techniques also known as Machine Learning. For example, in our case study, we have to predict future highs and lows of payment services. These insights could be useful to optimize the service offered (e.g. high availability and low latency) or prevent the loss of customers (customer churn). To do this, we can implement a Random Forest Regression algorithm that allows us to predict the highs and lows depending on certain variables, such as the number of services subscribed, service usage time or other typologies related to the user (e.g. level of studies, purchasing power).

Enrichment processes can consume data from i) the raw repository, ii) a streaming PEM acquisition process or iii) preprocessed data stored in one of the analytical repositories (REP) of our Big Data pipeline. Enrichment processes often require pre-processing sources for cleaning, normalization, integration, elimination of duplicates or treatment of null values and outliers. This type of processing must be implemented by existing or new PETL processes, using the implementation techniques and tools we have already discussed previously.

PEN processes that pursue the addition and integration of new data sources must also be implemented with the same type of tools as PETL processes, using again the requirements identified in the previous sub-section for the evaluation and selection of the most appropriate data transformation tools. However, PEN processes based on the use of Machine Learning techniques require specific tools to implement them. According to existing research [7] that identifies specific requirements for Machine Learning (ML) tools in Big Data, we have specified the following requirements for the evaluation of the most suitable tools for the implementation of each PEN process:

- **Volume:** As in other cases, the volume of data (R1) that can be processed by ML tools is very important [7], especially with regards to the processing opportunity (R3). Technologies such as Spark, Flink or H2O may run a wide variety of Machine Learning libraries such as MLib, Mahout, H2O or even R language, with the efficiency and scalability of distributed processing of a Hadoop environment. However, classic technologies such as R Studio or Python (e.g. with ML libraries such as Pandas or Scikit) are limited to the resources of a single machine (vertical scaling), as they do not implement distributed execution.
- **Velocity:** As with other types of processes considered in our methodology, it is important that PEN processes meet the processing opportunity constraints (R3). Tools such as Spark or Flink enable the application of previ-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3039455, IEEE Access

TABLE 10. Evaluation of ML processing technologies to implement PEN

Tool	Libraries	Volume R1	Velocity R3	R4	Variety R7
Map Reduce	Mahout	High	Batch	Batch	Medium
Spark	MLib, R, Mahout, H2O (Sparking Water)	High	Batch and Near Real-Time	Interactive	High
Flink	FlinkML	High	Batch and Real-Time	Interactive	Low
Kafka Streams	H2O, Tensor Flow	High	Real-Time	Interactive	Medium
H2O	H2O	Medium	Batch	Semi Interactive	Low
R Studio	R, H2O	Low	Batch	Semi Interactive	High
Python	Scikit, Pandas, ...	Low	Batch	Batch	High
Tensor Flow	Tensor Flow	Medium	Batch	Batch	Low

ously created ML models on real-time data and even the creation (training) of such models. However, tools such as RStudio, H2O or Map Reduce (Mahout) do not have the possibility to carry out ML processing in streaming. On the other hand, we must consider query latency (R4) when applying the previously created analytical models to the input data. Their response times may affect interactivity constraints, thus violating the application requirements.

- **Variety:** As in [7], we identified the variety and complexity of analysis models (R7) as one of the most important factors for ML tools selection. In practice, this coverage can be measured through a variety of libraries and algorithms supported by each technology. For example, Spark technology has a very wide coverage [7] because it supports a number of libraries such as MLib, R language, H2O or Mahout, essentially including almost all the existing ML algorithms.

Based on the above requirements, we have evaluated some of the existing ML technologies, especially those with suitable features for Big Data processing [7]. The result is shown in Table 10.

Applying the proposed requirements to evaluate the PEN_1 process of our e-commerce case study, we can select the most suitable tool for its implementation using Table 10 as a reference. The result of this process is shown in Table 11.

After a first evaluation of PEN_1 process, Spark and Flink were possible candidate technologies to satisfy R1, R3 and R4. However, the data scientist of the e-commerce company identified the Random Forest and Gradient-Boosted Trees Regression algorithms as the most suitable for the implementation of the prediction process. Since these algorithms are not available in the Flink ML, Spark with MLib was the technology selected.

At the end of this phase, we have already selected the

TABLE 11. Evaluation of the enrichment processes and tool selection

Process	Volume R1	Velocity R3	R4	Variety R7	Selected Tool
PEN_1	Medium	Batch	Interactive	High	Spark

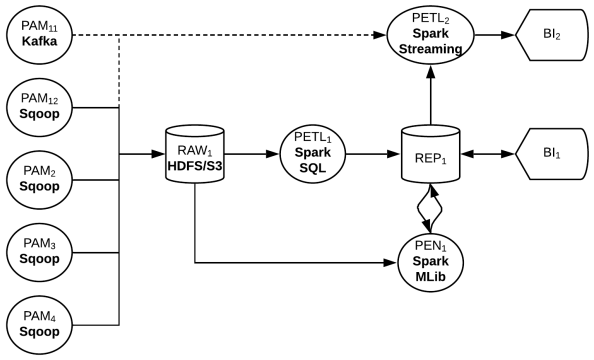

FIGURE 5. Big Data Pipeline for the e-commerce case study after the application of phase 3.

technologies for the implementation of the different PETL and PEN processes of our application case. The last step is to refine the current graph of our Big Data Pipeline to label each PETL and PEN process with the selected technology. The Fig. 5 shows the updated graph at the end of this phase for the e-commerce case study.

F. PHASE 4. ANALYTICAL REPOSITORIES IMPLEMENTATION

At this point, we have already defined the processes and tools of the architecture to support data sources acquisition, integration and pre-processing. The goal of this fourth phase is to select the most suitable technology for non-raw data repositories that provide analytical capabilities (REP).

Below, we specify the requirements from Table 1 that have more importance [5], [30], [31] in the selection of the technology for the analytical repository, also called Big Data Warehouse [16]:

- **Volume:** We have to evaluate the data size (R1) supported by a tool in relation to the required processing opportunity (R3) and query latency (R4). A Big Data repository should be able to scale up to hundreds of petabytes if necessary [5]. Also, it is important to evaluate the number of rows (R2) in the analytical MD models tables, since we have joins between these tables. Join operations are extremely costly, especially when we are dealing with tables of billions of rows. Similarly, the number of rows affects the performance of queries with aggregation operations, where columnar storages such as Hive (using ORC or Parquet formats), Kylin (using HBase) or Vertica, can benefit query performance and also storage space optimization.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3039455, IEEE Access

- **Velocity:** With regards to the velocity, we must evaluate if the technology enables adding or refreshing data from analytical MD models within the processing opportunity constraint (R3). For example, Apache Kylin allows incremental loading and refreshing of batch and near real-time data through micro-batch processes. However, Apache Druid allows data loading and refreshing in real-time. In addition, when a repository aims to support some BI tool or application, providing an interactive analytical query execution time (R4) is very important to improve productivity in the extraction of knowledge by end users. Technologies such as Apache Kylin support sub-second query latency on MD analytical models with tables of billions of rows.
- **Variety:** The complexity of the analytical models supported (R7) varies from tool to tool. For example, Apache Kylin and Hive allow to implement multidimensional analytical models (MD) composed by multiple tables (e.g. star or snowflake schemas), as the model of the Fig. 4. However, tools such as Druid require all data to be combined (denormalized) in a single table, thus allowing less complete and hard to maintain models. Moreover, we have to evaluate the types of data structure supported (R6). Tools such as Spark allow semi-interactive queries on unstructured data, but others such as Kylin or Vertica require pre-structured data.

We have used the above requirements to evaluate some of the most current Big Data repository technologies [5], [30], [31]. We have only evaluated technologies that allow data to be stored in tables and that support SQL language. Both these features facilitate the implementation and maintenance of MD data models for the Big Data Warehouse implementation. In addition, since SQL is a well-standardized data query and manipulation language, the know-how required for the use of SQL-based tools is reduced compared to other tools.

The result of the evaluation of the repository tools against the above defined non-functional requirements is shown in Table 12. For a more accurate assessment of performance-related requirements (R3 and R4) and storage capacity (R1 and R2) in relation to performance, we recommend applying a benchmark such as those of Transaction Processing Performance Council (TPC) [35], e.g. TPC-DS or TPC-H, which are widely used and standardized.

Once we have evaluated the available technologies for analytical repositories, we can apply these results for the selection of the technology of each analytical repository in our application scenario. For each repository (REP) we have to evaluate the identified requirements (R1,R2,R3,R4,R6,R7) and, based on their ratings, we have to select the most suitable technologies using Table 12 as a reference.

In our e-commerce case study, we have to implement the REP2 analytical repository to support the following BI applications and processes:

- BI_1**:** Interactive reporting/dashboarding system with a targeted query latency of less than 5 seconds and never more than 10 seconds.

TABLE 12. Evaluation of technologies to implement the analytical repositories (REP)

Tool	Volume R1	R2	Velocity R3	R4	Variety R6	R7
Apache Kylin	High	High	Batch and NRT	Interactive	Struct. & Semi Struct.	High
Apache Hive (Hive on Tez)	High	High	Batch	Semi Interactive	Struct. & Semi Struct.	High
Apache Impala	High	High	Batch	Semi Interactive	Struct.	High
Vertica	High	High	Batch and NRT	Interactive	Struct. & Semi Struct.	High
Spark (SQL)	High	High	Batch	Semi Interactive	Struct. & Semi Struct.	High
Apache Druid	High	High	Batch and RT	Interactive	Struct. & Semi Struct.	Low

TABLE 13. Evaluation of the technology for the analytical repositories implementation

Rep.	Volume R1	R2	Velocity R3	R4	Variety R6	R7	Selected Tool
REP_1	Medium	High	NRT	Interactive	Structured	High	Kylin (+Hive)

- $PETL_2$**:** Process for the analysis of live customers, subscriptions and unsubscriptions in real-time. This process requires loading historical and other context data from the REP_1 repository to enable real-time integration and aggregation with the new data received through $PETL_1$.
- PEN_1**:** Forecasting process of subscriptions and unsubscriptions with a daily retraining model. For the training and retraining of the predictive models, this process combines data obtained directly from the Raw_1 repository with data from the REP_1 repository, previously transformed by the $PETL_1$ process. In addition, the results of the predictions generated must be persisted in the MD model implemented in REP_1 in order to enable its consultation by end users.

Considering the above specifications, we have performed the evaluation of the REP_1 requirements (R1,R2,R3,R4,R6,R7) using Table 12 as a reference for the selection of the most suitable repository technology. The result of this process is shown in Table 13.

After a first evaluation, the candidate technologies were Vertica or the combination of Kylin and Hive. Both technologies enable the implementation of multidimensional analytical data models, supporting SQL query language enabling interactive queries on petabyte sizes and tables of billions of rows. However, although Vertica supports integration with Hadoop, the deployment of an external Hadoop tool could add more complexity (know-how) to the Big Data Pipeline. For these reasons, we finally opted for the combination of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3039455, IEEE Access

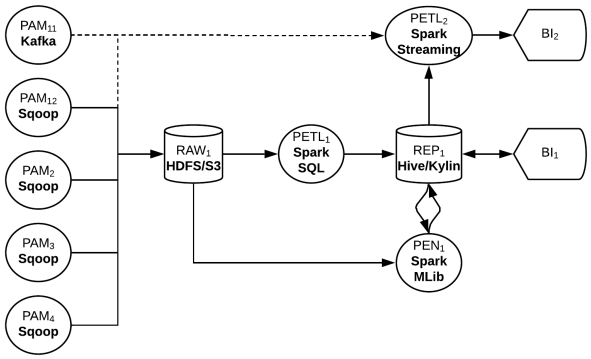

FIGURE 6. Big Data Pipeline for the e-commerce case study after the application of phase 4.

Apache Hive and Kylin.

The reason for using both Kylin and Hive is that Kylin easily integrates with Hive enabling row-level data updates and other features that complement Apache Kylin. Thus, Hive supports SQL queries for data transformation, updating and export ($PETL_1$, $PETL_2$ and PEN_1 processes) while Kylin supports analytical type SQL queries with sub-second latency required by BI_1.

Finally, once the technologies for the analytical repositories have been selected, we have to refine our Big Data Pipeline by tagging the REP nodes with the selected technologies. In Fig. 6, we show the Big Data Pipeline diagram at the end of this phase.

G. PHASE 5. DATA VISUALIZATION AND ANALYSIS

At this point, we have already identified all the processes and repositories in our Big Data Pipeline as well as the technology necessary for their implementation. In this last phase 5, we have to add the necessary technologies for the implementation of the BI applications required in the specific application we are addressing. These BI applications are usually graphs, dashboards, reports or multidimensional tables (OLAP views). They can be implemented using a Business Intelligence tool or, if none of the BI tools on the market covers all the requirements of our target application, by creating an Ad-Hoc application.

As we did in previous phases, we have identified the most important requirements of Table 1 to evaluate BI technologies for using them in our Big Data Pipeline:

- **Volume:** Evaluate the size of the data (R1) in relation to the scalability of the data sources supported by the BI tool. The direct connection implies that the data to be loaded in the dashboards are obtained through interactive queries to the repository. An existing alternative to direct connections are the tools that require data to be loaded (copied) into the BI system memory (In-Memory), which sometimes forces us to work with subsets of data or aggregated data. It is common to find BI tools that support both architectures, such as Power BI or Microstrategy. In these cases, we will evaluate R1 taking into account the most scalable mode of operation. As for the volume of rows (R2) of the tables of the analytical models, we will evaluate the BI tools with respect to the performance of rendering the data in the data in graphs or tables.
- **Velocity:** We have to evaluate both the processing opportunity (R3) as well as the query latency (R4) supported by the tool. Most BI tools use a batch approach (R3) with support for connecting to repositories that enable interactive queries (R4). In addition to batch and interactive processing, there are other tools like Microstrategy that enable the update of visualizations in real-time from repositories such as Druid, or from real-time acquisition systems such as Kafka.
- **Variety:** We have to evaluate the level of structure (R6) that the tool is capable of processing. For example, tools such as Superset only support structured data sources. Others such as Power BI allow you to connect directly to raw repositories (e.g. HDFS) for loading less structured data and even preprocessing them on the fly (light PETL processes). On the other hand, we have to evaluate the strengths of the metadata layer, used by the BI tool to represent complex analytical models (R7) over data origins. In this regard, we will value features such as the definition of new metrics calculated from data sources.

TABLE 14. Evaluation of BI technologies to implement BI applications (BI)

Tool	Volume R1	R2	Velocity R3	R4	Variety R6	R7
Power BI	High	High	Batch & RT	Interactive	All types	High
Micro-strategy	High	High	Batch & RT	Interactive	All types	High
Superset	High	High	Batch & NRT	Interactive	Struct.	Low
Tableau	High	High	Batch	Interactive	Struct. & Semi Struct.	Med
Pentaho BA	High	Low	Batch	Interactive	Struct. & Semi Struct.	High

Considering the above requirements, we have conducted an evaluation of some of the most valued BI tools [34] that offer features for Big Data processing. In Table 14 we show the result of this evaluation.

Once we have evaluated the possible technologies for the implementation of BI applications, we can apply the results to the selection of the technology for our particular application.

In our e-commerce case study we have identified two BI applications with different requirements:

- BI_1**:** Interactive reporting/dashboarding system with an objective query latency of less than 5 seconds and never more than 10 seconds.
- BI_2**:** Dashboard for the analysis of active customers, subscriptions and un-subscriptions in real-time (refreshing time <60 seconds).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3039455, IEEE Access

TABLE 15. Evaluation of the technology for the BI applications implementation

BI App	Volume R1	R2	Velocity R3	R4	Variety R6	R7	Selected Tool
BI_1	Medium	High	Near Real Time	Interac-tive	Structured	High	Micro-strategy
BI_2	Low	Low	Real Time	Interac-tive	Semi Structured	Low	Micro-strategy

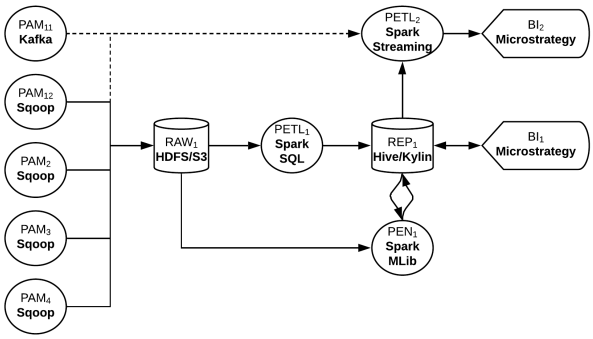

FIGURE 7. Big Data Pipeline generated for the e-commerce case study at the end of the methodology application.

Considering the case study specifications, we conducted the evaluation of the BI processes requirements (R1,R2,R3,R4,R6,R7) using Table 14 as a reference for the selection of the most suitable tools to implement them. The result of this process is shown in Table 15.

Regarding BI_1, we have discarded Power BI because Kylin in its open source version has no direct connection to Power BI and can only work in import mode, that requires copying data to the Microsoft cloud. This prevents the fulfillment of some requirements such as R3 and R1. We have also discarded Pentaho for the graphical performance (R2) and Superset for the low coverage (R7) of the metadata layer. As a result, only Microstrategy and Tableau were positioned as suitable tools to support the BI_1 application.

For the implementation of BI_2, we needed a BI tool that supported the connection with the PETL2 process for real-time analysis (R3). Spark technology allows to expose the processed data in real-time through Spark SQL using distributed In-Memory approach. The BI tool can connect to Spark SQL using J/ODBC connectors. Therefore, we need a technology that supports direct connection to Spark and it also has good refreshing rates for visuals. In this case Microstrategy and Power BI were suitable alternatives, but we opted for Microstrategy as it can be used to implement all BI applications in our case study.

Finally, we have carried out the labeling of the BI applications with the technology selected on the Big Data Pipeline graph. The Big Data Pipeline generated by the application of our methodology is shown in Fig. 7.

IV. BIG DATA PIPELINE EVALUATION

Through the application of the proposed methodology, we have generated a Big Data Pipeline that takes into account the non-functional requirements (derived from 5Vs of Big Data) of the target analytical applications. Next, we have to deploy the Big Data Pipeline using the most appropriate hardware. However, the selection and sizing of hardware is a complex task that requires extensive analysis that is beyond of the scope of this research.

Once deployed, the Big Data Pipeline should meet the constraints identified for the Big Data scenario it is intended to cover. Since this may not occur, we have to evaluate the Big Data Pipeline and if any part of it does not meet the requirements, we must iterate over the application of the methodology trying to refine its design to accomplish them.

To validate the Big Data Pipeline implemented, we consider critical the requirements of processing opportunity (R3), query latency (R4) and data quality (R8). The fulfillment of the rest of the requirements (R1,R2,R5,R6,R7) is directly or indirectly related to R3 and R4, thus we are not obviating them for the evaluation.

Therefore, the evaluation process we propose is based on the comparison of real executions values against the constraints defined for the requirements R3, R4 and R8. This process differs slightly depending on the requirement evaluated:

- **Processing Opportunity (R3):** For each process, repository, or BI tool, we calculate the average execution time from actual execution time samples. Then, to obtain the total time, we will add the average times of the sequentially executed processes. Often some processes such as data acquisition and movement (PAM) are executed in parallel. In this case, we have to use the time of the longest path.
- **Query latency (R4):** Analytical queries running in repositories are generated by user interaction with BI applications such as dashboards or reports. We will measure both the average times of execution of the queries in the repositories, as well as the average times of loading and rendering the data in the graphical elements of the BI application.
- **Data Quality (R8):** We will evaluate the quality of the data processed and stored in the repositories in terms of generation of duplicated records and loss of records. To carry out this task, we will calculate the difference between the number of actual records in each analytical repository (REP) and the number of expected records. Then, we will calculate the % of deviation (error) in order to evaluate the fulfillment of this requirement.

In the event that any of the requirements R3 or R4 have not been satisfied, the following steps must be followed:

1) Scale the hardware vertically, adding processors or RAM memory to the machines, or horizontally, adding hosts to the Hadoop cluster. After scaling resources, repeat the evaluation process.

2) If hardware scaling is not improving the results, iterate on the application of the methodology in the phases that address the processes that failed the evaluation, trying to refine the design of the Big Data Pipeline. After each iteration, repeat the evaluation process.

As for the R8 requirement, if it is not met, we have to iterate in the application of the methodology by reviewing the capabilities of the tools chosen, as well as the quality of the processes implemented (PAM, PETL, PEN). We have to make sure that no duplicates are generated, or records are lost above the % defined for the R8 requirement in the application.

A. E-COMMERCE USE CASE EVALUATION

By the application of the proposed methodology, we generated the Big Data Pipeline shown in Fig. 7. Next we detail the software the hardware chosen for its deployment:

- **Hadoop Cluster:** Hortonwork Data Platform (HDP) 3.1 distribution which includes Hadoop (Yarn and HDFS), Kafka, Sqoop, Spark, Hive and HBase (required by Kylin). Apache Kylin 2.6.6 is not included with the distribution, but it has been installed on one of the Hadoop cluster hosts. The hardware used for its deployment is 6 hosts m5.4xlarge in the Amazon AWS cloud, with 16 vCores and 64 Gb of RAM each.
- **MicroStrategy:** Including Intelligence Server, Web and Messaging Services. In this case, we have used a single r5.xlarge instance with 4 vCores and 64 Gb of RAM.

Once the previous architecture has been deployed, all the identified processes have been implemented and their real execution times have been measured. Although we will not mention the implementation details of the processes due to its extension, we highlight that they have been implemented taking full advantage of the parallel and distributed execution of the chosen tools and the Hadoop cluster. A code-level optimization of all the processes developed, such as PETL and PEN instances, has been carried out.

First, we will conduct the A1 application evaluation aimed to the interactive reporting of metrics such as active customers, subscriptions, unsubscriptions, impressions or clicks. For the e-commerce company it was critical that the incremental processing time (R3) of new data and updates was less than or equal to 15 min. Also critical was the query latency time (R4) when interacting with dashboards and reports: It has to be less than 5 seconds and never more than 10 seconds. Finally, for A1 no margin of error was allowed in the data due to the generation of duplicates or the loss of data (R8).

In Table 16 we show the evaluation of the processing opportunity taking into account all the processing required by application A1. The data acquisition and movement processes (PAM) are executed in parallel, so we have only taken into account the average time of the longest PAM process. The rest of the processes are executed sequentially, as they depend on the data generated by the previous process.

The query latency (R4) depends on the REP_1 repository and the BI_1 tool. The real value for REP_1 is the average

TABLE 16. Evaluation of R3 for the A1 application of the e-commerce case study

Process	Target Value	Real Value	Variation	Compliance
$PAM_{12,2,3,4}$	NRT (<=15 min)	3.3 min	11.7	Yes
$PETL_1$	NRT (<=15 min)	5.1 min	9.9	Yes
REP_1	NRT (<=15 min)	4.2 min	10.8	Yes
BI_1	NRT (<=15 min)	<1 sec	<1 sec	Yes
Total	NRT (<=15 min)	12.6 min	2.4 min	Yes

TABLE 17. Evaluation of R4 for the A1 application of the e-commerce case study

Process	Target Value	Real Value	Variation	Compliance
REP_1	Interactive (<5 sec)	0.29 sec	4.70 sec	Yes
BI_1	Interactive (<5 sec)	1.23 sec	3.77 sec	Yes
Total	Interactive (<5 sec)	1.53 sec	3.47 sec	Yes

of 113,694 queries monitored over 3 months of the interaction of end users with the BI_1 application, resulting in an average latency time of 296.6 milliseconds. Regarding the rendering of this data in the dashboards and reports created in Microstrategy, the average execution time without counting the execution time of the queries was 1.23 seconds. Results of R4 evaluation for the A1 application are shown in Table 17.

Finally, for the evaluation of R8, we examine if there are duplicates or missing data, by running record count queries against the R2 repository and comparing the results with queries made directly on the data sources. The result of this verification indicates that there are no duplicates or missing data.

As a result, the Big Data Pipeline generated by the application of the proposed methodology is validated for application A1. Similarly, we have applied the evaluation to the remaining e-commerce case study applications A2 and A3, achieving the same satisfactory result.

B. IOT USE CASE EVALUATION

The methodology proposed in this research is an evolution of our previous version of the methodology [16], which was applied to a case study of IoT. We have also applied the new version of the methodology to the IoT case, in order to validate its use against a second use case quite different from the e-commerce case.

The IoT case study aimed at providing real-time analysis of power consumption and generation data obtained from 52 Smart Metters devices installed in homes around the world. These devices were connected to an open IoT network that enables the distribution of these data in real-time through web services and JSON file format of semi-structured type. Moreover, there are some small differences in the data model of these 52 sources such as the number of fields, field names or used units of measure.

Considering the above specifications of this unique A1 application, we have generated the Big Data Pipeline shown

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3039455, IEEE Access

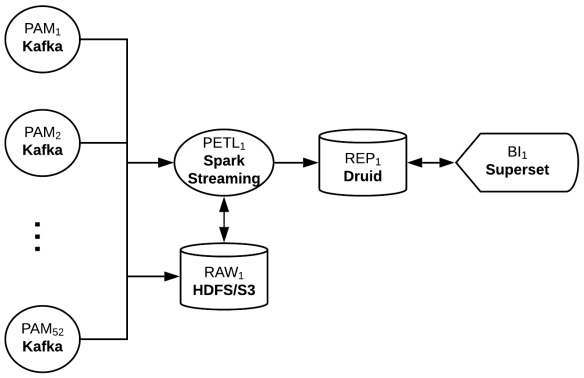

FIGURE 8. Big Data Pipeline generated for the IoT case study.

TABLE 18. Evaluation of R3 for the A1 application of the IoT case study

Process	Target Value	Real Value	Variation	Compliance
$PAM_{[1,52]}$	RT (<=1 min)	<1 sec	<1 sec	Yes
$PETL_1$	RT (<= 1 min)	20 sec	40 sec	Yes
REP_1	RT (<= 1 min)	<1 sec	58 sec	Yes
BI_1	RT (<= 1 min)	<1 sec	<1 sec	Yes
Total	RT (<= 1 min)	22 sec	38 sec	Yes

in the Fig. 8 using the new version of the proposed methodology.

In the Big Data Pipeline we have selected Kafka as the most suitable tool for the acquisition of the 52 data sources in real-time. In addition, we have considered the need to use a raw repository (Raw_1) that stores the data in both HDFS and S3, to store the sources in their original format and thus be able to explore them using the Spark tool in the phase 3. The result of the data exploration is an MD analytical model that allows for the integration of the 52 data sources and supports the required analytical queries. In order to implement the integration of the data sources and to loading this MD model, it is necessary to implement the $PETL_1$ process in real-time (R3, <= 60s) using the Spark tool.

As for the repository REP_1, we have selected the Apache Druid tool that enables the required interactive query latency (R4) over the data stored using one MD model. At the same time, Druid enables real-time data loading in times ranging from milliseconds to a few seconds. Then, the data is inserted into Druid through the Spark Streaming connection. Finally, Apache Superset has been the tool selected in the phase 5 for the implementation of the BI application.

Once the Big Data Pipeline was designed, we also deployed it using the Hadoop Hortonworks HDP 3.1 distribution on a cluster of 6 m5.4xlarge hosts in the Amazon AWS cloud. After the implementation of the different processes that comprise our Big Data Pipeline, we have carried out the evaluation of this pipeline following the proposed steps. The results of this evaluation for R3 (Processing Opportunity) and query latency (R4) are shown in Tables 18 and 19.

TABLE 19. Evaluation of R4 for the A1 application of the IoT case study

Process	Target Value	Real Value	Variation	Compliance
REP_1	Interactive (<=5 sec)	1.6 sec	3.4 sec	Yes
BI_1	Interactive (<=5 sec)	2.4 sec	2.6 sec	Yes
Total	Interactive (<=5 sec)	4 sec	1 sec	Yes

As these tables show, both R3 and R4 compliance is achieved. Druid enables data from all 52 sources to be ingested into the MD model with an average of 233 milliseconds (1000 samples). Considering the average times of the remaining processes, the data is available for querying in about 22 seconds since the moment they are generated in the data sources. Then, the data is shown in dashboard created with Superset, which is refreshed every 4 s. As a result, end users or analysts can analyze the data only 26 seconds after they are generated.

Using these two case studies, we have successfully validated the Big Data Pipelines generated by our proposed methodology, both in e-commerce as well as in IoT environments, complying in both cases with all the non-functional requirements identified in Table 1.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have reviewed the state-of-the-art on methodological approaches for the generation of Big Data Pipelines, that is, architectures that support Big Data Analytics. As a result, we observed that most methodological proposals are based on the analysis of the requirements derived from the 5V's or Big Data features. However, as we have seen, current methodologies either do not guide the development of the Big Data architecture with enough detail, or are rather sector or application specific.

To overcome these limitations, we have proposed an iterative methodology for the definition of Big Data Pipelines based on the analysis of non-functional requirements derived from the 5V's of Big Data. These requirements have been identified by reviewing the features in current Big Data applications, with the aim of ensuring that the proposed methodology is applicable to any Big Data scenario.

Moreover, the iterative feature of our methodology systematizes the evaluation and redesign of the different parts of the generated architecture in case the requirements of the analytical applications are not met. This results in more efficient and complete architectures than with other non-iterative methods.

It is important to highlight that the proposed methodology is based on our previous work [16]. The new version of our methodology is an evolution of the original, focused on the generation of the Big Data Pipeline. In this new version, we provide more detail regarding the set of non-functional requirements to be considered as well as a graphical notation and an algorithm for automating the generation of the structure of the Big Data Pipeline from the analysis of these requirements. In addition, the tools and techniques evaluated

for each element of the Big Data Pipeline have been updated, taking into account novel and powerful tools such as Kylin or Druid.

In addition, we have evaluated the proposed methodology with a real case study from the e-commerce sector. Thanks to the application of the proposed methodology, a Big Data Pipeline has been implemented, using a mix of tools from the Hadoop environment combined with some external ones. The implementation of the Big Data Pipeline has allowed this company to analyze the hundreds of terabytes generated by its different business processes (e.g. customer acquisition, billing, marketing) for the extraction of useful knowledge (e.g. decision making in marketing campaigns) in an efficient and effective way. In addition, the methodology has also been applied again to the IoT case study presented in [16]. The result of the evaluation of the generated Big Data Pipeline has also been satisfactory, thus validating our proposed methodology using two case studies in well-differentiated fields of application.

However, despite our efforts to simplify and automate the Big Data Pipeline generation process, a significant manual effort and a certain Know-How of Big Data tools is still required. This way, as future work we propose the inclusion of databases or the definition of ontologies to store the features of existing Big Data tools (requirements derived from the 5V's) and the target applications of the case at hand. These databases would be used as input of the proposed algorithm for the generation of the Big Data Pipeline, facilitating its automation and reducing the Know-how required by IT professionals.

REFERENCES

[1] G. Park, S. Park, L. Khan, and L. Chung, "IRIS: A goal-oriented big data analytics framework on Spark for better business decisions," in *Proc. IEEE Int. Conference on Big Data and Smart Computing (BigComp)*, Jeju, 2017, pp. 76–83.

[2] N. Kozmina, L. Niedrite, and L. Zemnickis, "Information requirements for big data projects: A review of state-of-the-art approaches," in *Proc. Int. Baltic Conference on Databases and Information Systems*. Cham, Switzerland: Springer, 2018, pp. 73–89.

[3] C. A. Ardagna, V. Bellandi, P. Ceravolo, E. Damiani, M. Bezzi and C. Hebert, "A model-driven methodology for big data analytics-as-a-service," in *Proc. IEEE Int. Congress on Big Data (BigData Congress)*, Honolulu, HI, USA, 2017, pp. 105–112.

[4] S. Orenga-Roglá, and R. Chalmeta, "Methodology for the implementation of knowledge management systems 2.0," *Business & Information Systems Engineering*, vol. 61, no. 2, pp. 195–213, 2019.

[5] Y. Cardinale, S. Guehis, and M. Rukoz, "Classifying big data analytic approaches: A generic architecture," in *Proc. Int. Conference on Software Technologies*. Cham, Switzerland: Springer, Jul. 2017, pp. 268–295.

[6] T. White, Hadoop: The definitive guide. O'Reilly Media, Inc, June 2009.

[7] S. Landset, TM. Khoshgoftaar, A. N. Richter, and T. Hasanin, "A survey of open source tools for machine learning with big data in the Hadoop ecosystem," *Journal of Big Data*, vol. 2, no. 1, p. 24, Springer, 2015.

[8] A.Gupta, A. Deokar, L. Iyer, R. Sharda, and D. Schrader, "Big data & analytics for societal impact: Recent research and trends," *Information Systems Frontiers*, vol. 20, no. 2, pp. 185–194, Springer, 2018.

[9] I. Olaronke and O. Oluwaseun, "Big data in healthcare: Prospects, challenges and resolutions," in *Proc. Future Technologies Conference (FTC)*, San Francisco, CA, 2016, pp. 1152–1157.

[10] H. Daki, A. El-Hannani, A. Aqqal, A. Haidine, and A. Dahbi, "Big data management in smart grid: Concepts, requirements and implementation," *Journal of Big Data*, vol. 4, no. 1, pp. 1–19, Springer, 2017.

[11] B. Baesens, R. Bapna, J.R. Marsden, J. Vanthienen, and J. L. Zhao, "Transformational issues of big data and analytics in networked business," *MIS quarterly*, vol. 40, no. 4, pp. 807–818, Dec. 2016.

[12] F. Di Tria, E. Lefons, and F. Tangorra, "Evaluation of data warehouse design methodologies in the context of big data," in *International Conference on Big Data Analytics and Knowledge (DaWaK)*, in Lecture Notes in Computer Science, vol. 10440, Cham, Springer, Aug. 2017, pp. 3–18.

[13] E. A. McGlynn, "Big data in health and healthcare: Hopes and fears for the future," in *Perspectives on Complex Global Challenges: Education, Energy, Healthcare, Security and Resilience*, 2016, pp. 113–115.

[14] X. Zhao, "A study on the applications of big data in cross-border e-commerce," in *Proc. IEEE 15th International Conference on e-Business Engineering (ICEBE)*, Xi'an, 2018, pp. 280–284.

[15] H. Eridaputra, B. Hendradjaya, and W. D. Sunindyo, "Modeling the requirements for big data application using goal oriented approach," in *Proc. Int. Conference on Data and Software Engineering (ICODSE)*, Bandung, 2014, pp. 1–6.

[16] R. Tardío, A. Mate and J. Trujillo, "An iterative methodology for big data management, analysis and visualization," in *Proc. IEEE Int. Conference on Big Data (Big Data)*, Santa Clara, CA, 2015, pp. 545–550.

[17] T. Grosser, J. Bloemen, M. Mack, and J. Vitsenko, "Hadoop and data lakes: Use cases, benefits and limitations. BARC research study," Nov. 2016. [Online]. Available: http://barc-research.com/research/hadoop-and-data-lakes

[18] P. Russom, "TDWI best practices report. The Data Warehousing Institute," Oct. 2013. [Online]. Available: https://tdwi.org/research/2013/10/tdwi-best-practices-report-managing-big-data.aspx

[19] N. S. Godbole and J. P. Lamb, Making healthcare green: The role of cloud, green IT, and data science to reduce healthcare costs and combat climate change. Springer, 2018.

[20] V. Potdar, A. Chandan, S. Batool, and N. Patel, Big energy data management for smart grids—Issues, challenges and recent developments, in Smart Cities: Development and Governance Frameworks. Springer, 2018, pp. 177–205.

[21] S. Akter and S. F. Wamba, "Big data analytics in e-commerce: A systematic review and agenda for future research," *Electronic Markets*, vol. 26, no. 2, pp. 173–194, Springer, 2016.

[22] M. Poblet, E. García-Cuesta, and P. Casanovas, "Crowdsourcing roles, methods and tools for data-intensive disaster management," *Information Systems Frontiers*, vol. 20, no. 6, pp. 1363–1379, Springer, Dec. 2018.

[23] M. M. Rathore, A. Ahmad, A. Paul, W. Hong, and H. Seo, "Advanced computing model for geosocial media using big data analytics," *Multimedia Tools and Applications*, vol. 76, no. 23, pp. 24767–24787, Springer, 2017.

[24] M. Avvenuti, S. Cresci, F. Del Vigna, T. Fagni, and M. Tesconi , "CrisMap: a big data crisis mapping system based on damage detection and geoparsing," *Information Systems Frontiers*, vol. 20, no. 5, pp. 993–1011, Springer, 2018.

[25] S. Nadal, V. Herrero, O. Romero, A. Abelló, X. Franch, S. Vansummeren, and D. Valerio, "A software reference architecture for semantic-aware big data systems," *Information and Software Technology*, vol. 90, pp. 75–92, 2017.

[26] M. Y. Santos and C. Costa, "Data warehousing in big data: from multidimensional to tabular data models," in *Proc. of the Ninth International C* Conference on Computer Science & Software Engineering*, New York, NY, USA, 2016, pp. 51–60.

[27] R. Kimball and M. Ross, The data warehouse toolkit: The definitive guide to dimensional modeling. John Wiley & Sons, 2013.

[28] S. Luján-Mora, J. Trujillo, and IY. Song, "A UML profile for multidimensional modeling in data warehouses," *Data & Knowledge Engineering*, vol. 59, no. 3, pp. 725–769, 2006.

[29] J. N. Mazón, J. Pardillo, and J. Trujillo, "A model-driven goal-oriented requirement engineering approach for data warehouses," in *Proc. of the Int. Conference on Conceptual Modeling (ER)*, Auckland, New Zealand, 2007, pp. 255–264.

[30] M.S. Wiewiórka, D. P. Wysakowicz, M. J. Okoniewski, and T. Gambin, "Benchmarking distributed data warehouse solutions for storing genomic variant information," *Database*, vol. 2017, 2017.

[31] M. Rodrigues, M. Y. Santos, and J. Bernardino, "Big data processing tools: An experimental performance evaluation," *WIREs Data Mining Knowl Discovery*, vol. 9, no. 2, 2019.

[32] E. Damiani, C. Ardagna, P. Ceravolo, and N. Scarabottolo, "Toward model-based big data-as-a-service: The TOREADOR Approach,"

in*European Conference on Advances in Databases and Information Systems (ADBIS)*, Cham, Springer, pp. 3–9, 2017.

[33] J. Lin, "The Lambda and the Kappa," in *IEEE Internet Computing*, vol. 21, no. 5, 2017, pp. 60–66.

[34] C. Howson, R. L. Sallam, J. L. Richardson, J. Tapadinhas, C. J. Idoine, and A. Woodward, "Magic quadrant for analytics and business intelligence platforms," Gartner, Aug. 2018.

[35] M. Poess, R. Othayoth Nambiar, and D. Walrath, "Why you should run TPC-DS: a workload analysis," in *Proc. 33rd international conference on Very large data bases (VLDB)*, Vienna, Austria, 2007, pp. 1138–1149.

[36] Z. Zhou, C. Gao, C. Xu, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Social big-data-based content dissemination in internet of vehicles," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 768–777, 2018.

[37] Z. Zhou, H. Yu, C. Xu, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Dependable content distribution in D2D-based cooperative vehicular networks: A big data-integrated coalition game approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 953–964, 2018.

ROBERTO TARDÍO was born in Madrid, Spain, in 1986. He received the degree in Computer Science Engineering from the University of Alicante, Alicante, Spain, in 2013. From the same university, he obtained a Msc in Computer Science Technology in 2014. He joined Lucentia in 2015, a research group associated to the Department of Software and Computing Systems of the University of Alicante. In this group, he carries out his studies of PhD, focused on in Big Data techniques and technologies.

He currently works as Head of Big Data for the consulting firm Stratebi Business Intelligence, based in Madrid, (Spain). For this company, he manages projects and equipment for the construction of Big Data platforms in large companies. In addition, he oversees the management of R&D in this company, succeeding in the incorporation of the newest Big Data technologies, such as Apache Kylin or Druid into current projects.

Mr Tardio conducts his current research in the following fields: Big Data architectures and Methods, Requirements Engineering, Data Modeling, OLAP tools and Database Benchmarking. In these areas, he has published several papers, among which we highlight two: "An iterative methodology for big data management, analysis and visualization" (IEEE Int. Conf. on Big Data, 2015) and "A novel multidimensional approach to integrate big data in business intelligence" (Journal of Database Management, 2015).

ALEJANDRO MATÉ holds a Computer Science Engineering degree from the University of Alicante since 2009, where he also obtained a Msc in Computer Science Technology in 2010 and a PhD in 2013. He was abroad for over 2 years as part of a Postdoc researcher position in Italy from 2014 to 2016, and worked for Lucentia Lab as Business Intelligence & Big Data architect as part of a Torres Quevedo grant from 2016 to 2017.

From 2017 to 2019 he was Assistant Professor at the Department of Software and Computing Systems. Since 2019, he has been Associate Professor at the University of Alicante. Throughout his career, he has collaborated with several research groups across the globe, most notably including the Requirements Engineering group led by John Mylopoulos at the University of Trento (Italy), and the Software Engineering group led by Eric Yu at the University of Toronto (Canada). His research has been mainly focused on BI and Analytics, ranging from the definition of strategic plans and Key Performance Indicators to the extraction of insights by means of Dashboards and algorithms. As a result, he has published over 50 papers related to BI and Analytics. Most of these papers are published in high impact international conferences (e.g. ER, CAiSE, RE) and JCR journals (Information Systems, Future Generations, Information & Software Technology).

Nevertheless, his career has not been limited to the research field. In the professional department, he has developed analytic systems and software for several national and international projects. Among these projects, we can find European Research Council grants (Lucretius) and large-scope national projects from private initiatives (LPS-Bigger). The novelty of the algorithms developed granted him the Best Demonstration Award at the IBM conference CASCON, in Canada. Currently, he is working on several projects related to eHealth and Internet of Things (IoT), combining real-time analytics and artificial intelligence.

JUAN TRUJILLO is a Full Professor at the University of Alicante, Dept. of Software and Computing Systems. Since he got his PhD in 2001, he has been leading the Business Intelligence and Big Data research in the department and has also been the founder and director of The Lucentia Research Group since 2008. His main research topics include Business Intelligence applications, Big Data processing and analytics, Data Warehouses, Decision Support Systems and Artificial Intelligence. He has advised 12 PhD students, and he is author of more than 200 conference paper, many of them in ERA A conferences, such as ER, UML, DAWAK or CAiSE, and more than 60 JCR papers, such as DKE, DSS, ISOFT, IS, or InfSci. He has been also co-editor of 11 special issues in multiple JCR journals, including DKE, DSS and CS&I. He has been PC-Chair in multiple international events, such as ER'18, ER'13, DOLAP'05, DAWAK'05-'06, and Juan C. Trujillo was Senior Editor of the Q1 JCR journal DSS (Decision Support Systems) until 2017.

It is also noteworthy the high impact of his publications in the field of BI, that have led him to become one of the most cited authors in the area, having papers with 152, 128 or 98 citations, positioned in the 3rd and 8th rank of the list of most downloaded papers in journals such as DKE. One of his papers appears as the most cited in the DKE journal during a 5 year period. More information and details on his research can be found at: https://dblp.uni-trier.de/pid/50/2311.html. Juan C. Trujillo is the most cited Researcher in the Technical School of Computer Science (EPS) in the UA and he is between the top 50 Researchers in Spain, considering all the disciplines within the Computer Science area, and he is within the 20 Top international researchers in his main areas such as conceptual modeling, data warehouses or Business Intelligence (Font: Google Scholar).

With regards to Technology Transfer, he owns 8 Intellectual Property Registers (IPR), and is co-founder of the Lucentia Lab S.L. Spin-off in April 2015, an EBT company participated by the University of Alicante. He has been Principal Investigator (PI) of a high number of National and Regional, and even International Research Projects.

Furthermore, he is a very active international researcher, and he is a member of several international associations, participating in the meetings of NESSI, PLANETIC, and BDVA (Big Data Value Association) among others. It is worth noting that this international activity and networking has materialized in the participation of several H2020 European projects, such as SAMNIC, SAFERPLAY (PI: Juan C. Trujillo), E4Children (PI: Juan C. Trujillo) and Lucretius (ERC Advanced Grant), where Juan C. Trujillo is an active researcher. Finally, he holds the international credential Project Management Professional (PMP®) for project management awarded by the prestigious Project Management Institute (PMI®).