Journal of Mobile Multimedia, Vol. 13, No. 1&2 (2017) 001–027

DATA INTELLIGENCE IN THE CONTEXT OF BIG DATA: A SURVEY

HICHAM MOAD SAFHI, BOUCHRA FRIKH
LTTI Lab, Sidi Mohamed Ben Abdellah University
Fez, Route d'imouzer,B.P. 2427 Morocco
h.m.safhi@gmail.com, bfrikh@yahoo.com

BADR HIRCHOUA, BRAHIM OUHBI
LM2I Lab, ENSAM, Moulay Ismail University
Meknes, Marjane II, B.P. 4024, Morocco
badr.hirchoua@gmail.com, ouhbib@yahoo.co.uk

ISMAIL KHALIL
Institute Telecooperation, Johannes Kepler University
Linz, Austria
ismail.khalil@jku.at

Mining Big Data is the capability of finding new useful information in complex massive datasets, that may be continuously changing and may have varied data types. Big data is helpful only when it is transformed into knowledge or useful information.
Data Intelligence is about transforming data into information, information into knowledge, and knowledge into value. It refers to the intelligent interaction with data in a rich, semantically meaningful ways, where data is used to learn and to obtain knowledge.

However, extracting valuable information from this data by following the classical Knowledge Discovery process reveals new previously unknown challenges, due to Big Data properties. These challenges have received a lot of attention in recent years, and still need more and more contribution and research. A large number of publications have yielded a plethora of proposed methods and algorithms.

In this paper, we provide a comprehensive literature review on Big Data current status. We present the Data Intelligence framework in the context of Big Data from data acquisition until insight extraction, we highlight its main issues, and identify its progress in both technological and algorithmic perspectives. We summarize and analyse relevant research papers in the field, collected from different scientific databases. This investigation will help researchers to understand the current status of Data Intelligence, discover new research opportunities, and gain information about this field.

Keywords: big data, data mining techniques, literature review, knowledge discovery.

1 Introduction

Data mining is the process of automatically discovering actionable information from datasets - Zekulin [1]. It requires the use of statistical methods and search algorithms to find structures, correlations, patterns and rules within data. It helps gaining knowledge, and getting understandable information in large databases [2] [3]. Generally, data mining techniques are utilized such as :

- Anomaly detection: Also known as outliers detection, it concerns finding the dissimilar object, i.e. an object in the data that deviates significantly from common pattern of the data. This object is considered as dissimilar to the data. Anomaly detection has been widely used in many applications, such as : data cleaning [4], fraud detection [5] ,intrusion detection [6] [7] , etc.
- Association rule learning: An other important data mining technique is association rules, which searches for interesting relation among variables in large database. The pattern reveals combinations of events that occur at the same time. Association rules are used in various applications: market basket analysis [8], medical diagnosis [9] [10], protein sequences [11] [12], image analysis [13] [14] and others.
- Clustering: clustering is the task of partitioning data into a set of clusters, in a way that similar objects are in the same cluster. It is an important technique used in many fields, including machine learning, pattern recognition, and image analysis.
- Classification: Classification used to identify class labels of a list of observations, and then classifies them to their categories, on the basis of a training set of data containing observations whose category membership is known.
- Regression: this technique used to estimate the relationships among variables by fitting an equation to the dataset. It can be used to model the relationship between one or more independent variables (attributes already known) and dependent variables (response variables or what we want to predict).

Data mining has become crucial in many fields including: health care, education, economy etc. Presently, many organizations are gathering data from different sources. Extracting knowledge from the collected data offers many new opportunities, and guides this organizations to make good decisions.

In fact, data mining is an important step in the knowledge discovery in database (KDD), which refers to the techniques applied to extract high-level knowledge from data. The KDD process consists of mainly five steps [15] : selection, preprocessing, transformation, data mining and interpretation/evaluation [Fig. 1]. In the first step, data sources that are susceptible to contain information are selected. The preprocessing consists of cleaning the target data. Transformation handles different data conversions and unification. After the data mining step, comes the interpretation of the pattern and the evaluation of the process.

To deal with the knowledge itself the knowledge management has been appeared. The European Committee for Standardizations official "Guide to Good Practice in Knowledge Management" defines Knowledge as : "... the combination of data and information, to which is added expert opinion, skills and experience, to result in a valuable asset which can be used to aid decision making".[16]

Managing any resource may be defined as doing what is necessary to get the most out of that resource. Therefore, at a very simple lev[1el, knowledge management may be defined as doing what is needed to get the most out of knowledge resources. Knowledge management can be defined as performing the activities involved in discovering, capturing, sharing, and applying knowledge so as to enhance, in a cost-effective fashion, the impact of knowledge on the units goal achievement.

The difference between Knowledge management (KM) and Business Intelligence (BI), is that KM incorporates knowledge capture, sharing, and application in addition to discov-

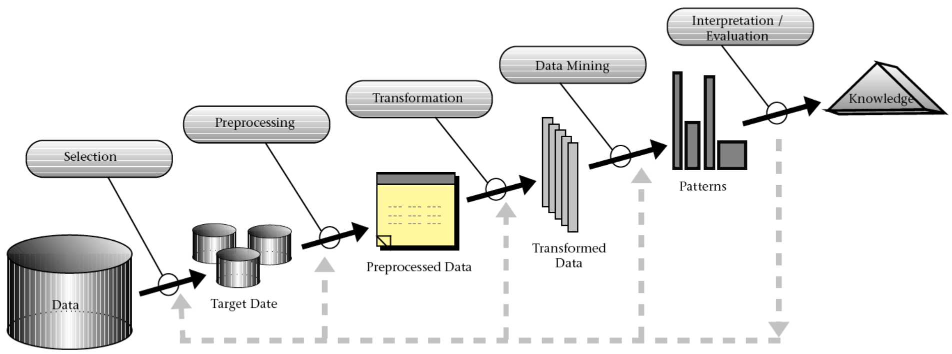

Fig. 1. Classical KDD process

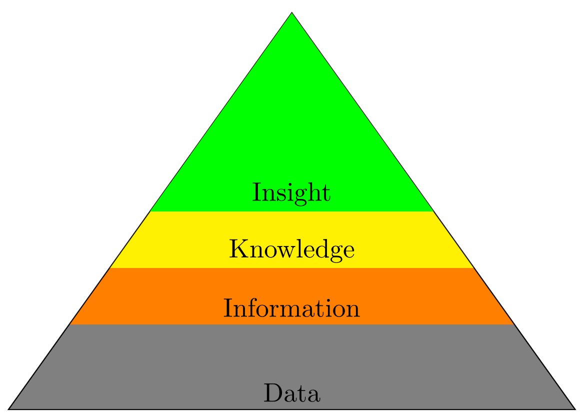

Fig. 2. from data to insight

ery.While BI focuses on data access, analysis, and presentation. The connection between BI and knowledge is limited to knowledge creation (by discovering patterns based on existing explicit data and information). Even in this respect, BI focuses directly on discovery of explicit knowledge whereas KM concerns discovery of both tacit and explicit knowledge. In other words, only explicit knowledge can directly result from BI, whereas KM is concerned with activities that produce both explicit and tacit knowledge.

Even though data mining techniques have reached a certain level of success, their direct application is still limited due to certain challenges. The rapid growth of data has actually generated a huge amount of it, that exceeds the ability of existing technology and techniques to process it. This kind of data is known as Big data, and it is not only characterized by volume, but also by other properties which are variety, velocity and value.

Extracting knowledge from big data is not always trivial. In 2006, [17] have addressed the biggest challenges in data mining, and among these complexities, some are related to complex, high dimensional, and high-speed data. Different approaches on how to overcome this challenges have been presented in recent years. We give through this paper an overview of the literature, by analyzing different proposed approaches in order to overcome big data's challenges in various knowledge discovery process steps.

In this paper, we discuss the following questions : What are the main challenges of extracting knowledge from Big Data? What tools and technologies are intended for mining Big Data, and what advantages do they offer? And what are the solutions provided by researchers in that field?

Mining Big Data had been covered under various survey, many existent papers have reviewed the challenges of big data mining in various fields, namely: [18] which gives an overview of big data analytics state of art, and its significant open problems in 2011, with a discussion about analyzing big multidimensional data.

[19] presents problems brought by big data in several domains, and explains the need of robust methods to infer noisy, complex and dependent big data. Authors in [20] provide a survey of machine learning techniques with a focus on big data. They present critical issues, and give some possible remedies with illustration of some learning methods that seems to be promising for surmounting big data problems. [21] gives a review of some recent approaches related to big data and data mining with their outcomes. [22] gives a systematic review of big data challenges among big data analytic life cycle, which can be seen as: data challenges, processing challenges, and management challenges. They have also analyzed some methods to overcome these challenges. [23] investigates big data mining frameworks and techniques for processing big graphs, which are practically considered very important in many applications. [24] have examined big data domain, with investigation of big data processing tools: their strength and their weaknesses. Techniques of mining social media, which is considered as an important source for big data [25], have been surveyed in [26]. Other reviews have addressed big data mining issues with a focus on a specific domain, including: education [27] [28], industry [29] [30], health [31] [32], text analysis [33], and disaster prediction [34].

Nevertheless, innovation in this field is nowadays occurring at high speed. Most of existing surveys track Big Data issues with domain dependent, or specific Big Data challenge. Thus, in this survey, we investigate Big Data issues from quiring the data until extracting knowledge, with domain independent. We track its latest development and status in different knowledge discovery and management steps.

2 Problems and challenges

Big data is a nascent concept that has uncertain origins [35], its exact definition is still debatable: it has at least 43 different definitions [36]. Literally, big data refers to information with massive volume and high dimensional space, it is the first notion that comes to mind when trying to define the term big data. However, volume is only one of several important characteristics of big data.

Generally speaking, big data definitions commonly include four challenges, known as 4 Vs. The 4 Vs are:

- Volume: refers to the size of data, it is often considered as the central feature of Big Data. Data storing has been facilitated by the decreasing price of disk storages. The generated data nowadays is reported in peta-bytes and zeta-bytes. Processing this massive data is challenging in the era of big data. In fact, most algorithms are designed to read data from memory, which is not always possible, because the capacity of hard drive storage far exceeds the one of memory.

- Variety: indicates the diversity of data types. Technological advances enabled the collect of varied data formats. In general, we can categorize data types into three groups:

  - Structured data: it has a defined format, and easy to be manipulated. eg. data in traditional databases.
  - Semi-structured data: it has a meta-data that helps to explain it. This data requires time and effort to be analyzed. eg. XML file.
  - Unstructured data: it does not have a specific schema. Also, it is the hardest data to be processed, because there is still a lack of technological solutions to automatically extract information out of it, eg. images, videos, etc.

  In reality, information does not reside only on structured data, as most of algorithms suppose, but also on other data types.

- Velocity: describes the rate at which data is being generated and in which it needs to be processed. Digital devices become more and more cheap, which has led to an unusual rate of data generation. Most systems need a real time response, which signify real time processing, that is difficult to achieve in big data.

- Value: considered as an essential aspect of big data. Value is the output of big data, and it is more important than the data itself. It is also a challenge to define the usefulness of analyzing big data, that should be clearly defined.

Many works conformed the definition of big data to their requirements, by adding new Vs to it. The additional Vs can stand for: Veracity [37], which describes confidentiality and integrity of data, it is about verifying data origins detecting noises and inconsistencies in data. Variability, stands for the difficulty of the dataset, such as the number of variables. Visibility highlights the need of a big picture about the data in order to make the decision.

3 A framework for Data Intelligence

We have proposed a framework to extract knowledge from big data, the steps of this framework are presented in figure 3. Our proposed framework consists of mainly two steps : Knowledge Discovery process, and Knowledge Management process.

Traditional data analysis are no longer adequate in extracting information from Big Data [38]. A simple way to reduce processing time when dealing with a problem with high complexity, is by using parallel processing techniques. It can be achieved using two methodologies: The first one consists of dividing data into N subsets, and applying the data mining algorithm on each subset separately, then we combine their outputs to obtain the final result. This methodology has been widely used, and known as ensemble methods.
The second is based on the MapReduce paradigm. Which is a popular programming model that simplifies parallel processing on a distributed system. MapReduce was published by Google in 2004 [39]. It consists of two steps: a map step, where input data is divided into independent key-value subsets that are executed in parallel. And a reduce step, where the intermediate values that are associated with the same key are merged to obtain the final result.

A comparison between these two models [40] showed that MapReduce methodology's performances are very stable and needs less computational costs to process big data in comparison with the distributed model, except for datasets with imbalanced class distribution.

MapReduce has gained much popularity, it has been implemented in many open source projects maintained by large companies.

Apache Hadoop [41] is a popular framework that implements MapReduce. Hadoop is an open source project that includes many other components designed to analyze big data, such as HDFS: a fault tolerance distributed file system, Hbase: a distributed NoSQL database, Hive: a data warehouse framework, Zookeeper: a coordination service for distributed applications, Mahout: a machine learning library, and other interesting packages destined to manage big data. However, Hadoop suffers from some drawbacks, the big one is that it stores each map/reduce temporal data into a file system, which makes the processing time very high in many applications. Hadoop's MapReduce is suitable for batch processing, but not for real time processing.

A new improved implementation of the MapReduce algorithm is Apache Spark [42]. Which provides the advantage of keeping the temporal data in memory, instead of storing it into the file system. This makes Spark's performances 100 times better than Hadoop, and makes it able to handle real-time data. Spark is good at data exploration, and it has some useful libraries such as MLlib for machine learning algorithms, and GraphX for graph processing.
Storm [43] is specially designed to process real-time data via computational graphs called 'topologies'. Storm does not have a library for machine learning, but it can go with other packages such as SAMOA [44].
Flink [45] is originated from the Stratosphere research project which began in TU Berlin [46]. It offers the capability for both batch and stream processing. Flink has a machine learning library: Flink-ML, and can also be used with SAMOA.
H2O [47] is another framework for parallel processing and big data analytics; it includes packages for machine learning, statistics, and evaluation tools. H2O's engine processes data completely in-memory using multiple execution methods, depending on what is best for the

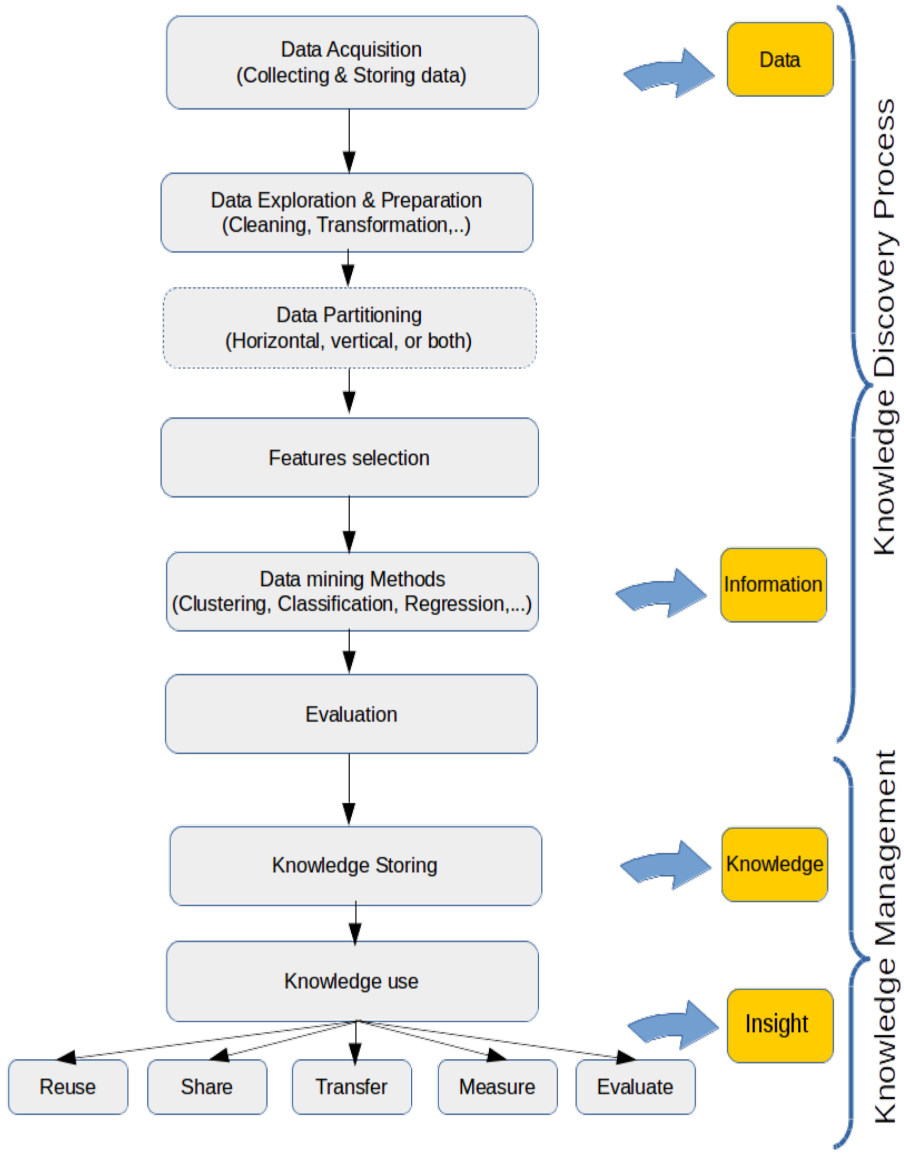

Fig. 3. our proposed framework

algorithm used. It also provides a web-interface that facilitates operations for analysts that may not have strong programming backgrounds.

We have made a comparative study between these technologies, as shown in Table 1.

Table 1. A comparative study between big data analytics platforms

	Hadoop	**Spark**	**Storm**	**Flink**	**H2O**
Current version	3.0	2.0	1.0.2	1.1.2	3.8.3.2
API Languages	Java, Python	Java, Scala, Python, R	Java, Scala, Python, Ruby, Clojure	Java, Scala, Python	Java, Scala, Python, R
Streaming support	No	Yes	Yes	Yes	Yes

3.1 Knowledge Discovery in the Context of Big Data

Despite the availability of big data infrastructures with the most popular machine learning algorithms, and the efficiency of the MapReduce paradigms to parallelize data processing, scientists still encounter many other challenges; on the one hand, MapReduce methodology cannot be directly applied to all algorithms. In fact, not all algorithms can be transformed to MapReduce jobs. Parallelization of those algorithms is not always a trivial task, and drives researchers to look up for more powerful solutions.
On the other hand, MapReduce considers that the mapped data is independent, and can be processed in parallel; which is not always true because some algorithms need to make interdependent predictions among data to take the final decision.
Researchers have been conducted to develop their specific algorithms, and refine existent ones to make them work in a big data context.

Knowledge discovery in databases process, i.e. the process of extracting knowledge from data using data mining techniques, is becoming increasingly important in a broad array of domains. Indeed, many organizations are currently relaying on data, and almost all of them involve the use of data mining.

However, many problems arises when applying the KDD process on big data. In this section we provide detailed description of this issues, on different KDD process steps, and the proposed solutions in literature.

3.1.1 Selection

In the selection step, two elements must be identified: the goal of the KDD, and the meta-data or the data that represents relevant prior knowledge. After that, the data set that may contain relevant information is selected or created. In general, data is often gathered from multiple sources and stored in a local sandbox. Suitable methods for storing massive unstructured data should be determined when collecting big data.

In the standpoint of volume, Big Data storing issues are resolved using distributed storage, in which a file is saved within a cluster of interconnected devices. In 2003 [48], Google published the paper about the distributed file system Google File System (GFS). It is a proprietary scalable file system developed in C/C++, that they used to scale their own search system using a cluster of commodity hardware. In 2010 [49], and based on the design of GFS, Yahoo and the open source community have developed the Hadoop Distributed File System (HDFS). It is based on Java, and is one of Apache top projects.

The distributed file systems are optimized to manage big files, and have the following properties:

- Transparency: Means that users are able to perform the same operations (file access, etc.) on the DFS in the same way as in local file systems.

- Fault tolerance: The design of the DFS takes into consideration that the system should not be stopped in case of partial failures (network problems, server failure, etc.).

- Scalability: Means that the system can handle large amounts of servers that are dynamically added to the system, without degrading performances.

The storage of a large file in HDFS, consists on dividing it into several blocks, each set of blocks are stored in a node. Reading a small part of the file directly from a node is impossible, because the data is stored in binary format. The user have to use the HDFS client, which reconstitutes the original file from all nodes. In this case, quiring data is practically demands resources, and time consuming, it is preferable to use databases.

Big data often demands higher performances in reading and writing data, which is considered as a challenge to traditional databases [50], they face many problems when dealing with unstructured data that frequently changes. Therefore, a new way of storing and manipulating data has emerged, known as NoSQL databases. NoSQL stands for Not Only SQL; Indeed, it presents many advantages: quick data reading/writing, support massive storage, easy scalability and low cost. The ACID properties (Atomicity, Consistency, Isolation and Durability) provided by RDBMSs are difficult constraints to guarantee in NoSQL databases. Instead of ACID, NoSQL provides BASE properties, which conforms the consistency model. That is:

a) Basically Available: the system is always available. Generally, data is divided and replicated, so that when a partition fails, it can be reconstructed again from replicas.

b) Soft state: the state of the system may not be always consistent, and it could change over time.

c) Eventual consistency: ensure the system's consistency at later times.

By implementing this concept, a full system failure is avoided, which guarantees a greater system availability [51].

NoSQL databases can be classified into 4 families, as shown in table 2. This families are:

- Key-Value Stores: Uses an identifier as key to locate values. The value may contain any type of data, from simple text to more complex data. This makes them fast and highly scalable. However, data is accessible only via the key, and can't be searched against values.

- Document Stores: As their name indicates, they are designed to store documents that are encoded in a standard data format like: XML, JSON or BSON. The value column contains semi-structured data, it may appear as multiple key-value pairs. The number and type of attributes can vary from a row to another.

- Column-Family Stores: also known as column oriented or wide-column. They store data that holds multiple attributes per key. It can store versioned blobs (byte stream) in one large table, and it can be easily scaled out.

- Graph/Triple Stores: They are very useful when the relationships within data is more interesting than data the itself. Searching in a graph databases is very fast, because recursive joins can be replaced by efficient traversals. However, they are not very scalable, especially when graphs don't fit into RAM, and they also use a specialized query languages.

However, storing all the data is often expensive, and still a big challenge. Not all the stored data contain the same amount of value.

Table 2. A comparative study between NoSQL databases categories

Family	**Description**	**Advantages**	**Limitations**	**Examples**
Key-Value	Keys used to locate the values.	Values may contain any type of data.	Data searches can only be via keys, not values.	S3; Berkley DB; DynamoDB; Redis
Document	Stores semi-structured documents.	Can keep document hierarchies.	Complex to implement	CouchDB (JSON); MongoDB (BSON); Couchbase
Column-Family	Holds multiple attributes per key.	Easily scale out, supports versioning	Can not query blob content.	Cassandra; HBase; Hypertable; Apache Accumulo; Bigtable
Graph / Triple Stores	Data stored in nodes +relationships +properties.	Fast search.	Poor scalability, needs specialized query languages.	Neo4j; Sones GraphDB; AllegroGraph; InfiniteGraph

3.1.2 Preprocessing

The preprocessing step consists of enhancing the quality of the dataset. This is achieved by the following tasks: a) supplementing missing attributes b) removing duplicate instances c) resolving data inconsistencies d) creating new attributes. Cleaning the data helps in increasing the data mining process performances, less noise in the data implies more efficient results.

However, when dealing with big data, it is very difficult to manually clean the dataset. One of the methods that automate this process is called features generation, it is the process of interrogating external data sources to get new valuable knowledge. Features generation can be used for: detecting outliers, giving meaning to variables, get domain-specific knowledge, and get additional significant attributes.

In a big data context, this step helps understanding the data, however, it could create a large amount of features, which makes the datasets quite larger.

3.1.3 Transformation

This step consists of analyzing the variables in term of correlation and importance. A crucial task in the transformation is features selection, it is used to identify relevant variables and the interaction between variables. This step helps in reducing the number of features by removing correlated and irrelevant ones. However, traditional algorithm's performances degrades when there is a big number of variables, which is the case of big data.

Classic feature selection algorithms can be classified into three categories: filters, wrappers, embedded and hybrid [52] [53].

Filters uses statistical search to rank the relevance of features. Wrappers are based on cross-validation, they measure the usefulness of feature subset to select the most useful features. Embedded methods are similar to wrappers, with the difference that the search is guided by a learning process. Hybrid methods combine multiple algorithms from the same category or not.

Features selection can also be formulated as a search problem [54]. In this case, methods are categorized into: exhaustive search, which evaluates the entire possible subsets. Heuristic search, that applies some techniques to guess the direction to the goal. And hybrid methods.

There has been research dedicated to this task, as it is considered an important step in big data, and it is also used in many fields : Microarray analysis, Image classification, Face recognition, Text classification, etc [55].

Authors in [56] proposed an algorithm : MR-EFS, that addresses the problem of features selecting using an evolutionary algorithm (EFS-CHC) for high-dimensional data. Their proposed algorithms was based on MapReduce paradigm. Input data is horizontally divided into subsets, and then, the algorithm is executed in a parallel way to select the features from each subset. After that, the results from each subset are combined to obtain the most appropriate features for the total database. This method has been tested with three classifiers, implemented in Apache Spark. Its performances have been measured in terms of the training time and classification accuracy using area under curve (AUC) metric.

As features selection is considered an NP-hard problem, many works proposed the use of approximative functions to find the suitable solution in an adequate time. [57] proposed HDBPSO algorithm for selecting features from gene expression data using Hamming distance as proximity measure based on binary PSO algorithm.

Authors in [58] proposed a method for distributing the feature selection algorithm. The data is distributed vertically: by features, two variants of data splitting are proposed: with and without ranking the original set of features. After that, a merging process is take in place to edit the selected features according to improvements in the classification accuracy. In most cases, the distribution model provides more performances and time efficiency in comparison with the centralized method.

[59] proposed an hybrid method that selects features in high dimensional datasets. To combine the advantages of both filter and wrapper, this algorithm applies filter-wrapper algorithms in two phases. First, the symmetrical uncertainty (SU) criterion is exploited to weight features in filter phase for discriminating the classes. Then, in wrapper phase, both FICA (fuzzy imperialist competitive algorithm) and IWSSr (Incremental Wrapper Subset Selection with replacement) in weighted feature space are executed to find relevant attributes.

To deal with imbalanced big data class distribution problem, [60] presented ROSEFW-RF algorithm, stands for : random oversampling and evolutionary feature weighting for random forest. Before building the model, this method combines multiple preprocessing stages, as its name suggest: random oversampling, it replicates randomly the instances of the minority class in order to balance the class distribution of the data. And evolutionary feature weighting, that selects the most significant features. All steps of this approach were built using MapReduce as parallelization paradigm.

Authors in [61] presented two hybrid features selection algorithms BDE-X$_{Rank}$ and BDE-X$_{Rankf}$, that combine a wrapper FS method based on a Binary Differential Evolution (BDE) algorithm with a rank-based filter FS method. In the first step, the features are sorted by using information gain filter. In the second step, a Binary DE-based wrapper method performs the search to find relevant ranked features. Difference between BDE-X$_{Rank}$ and BDE-X$_{Rankf}$ is that this last uses an additional fitness function that participate in selecting the adequate features.

[62] presents a study on the use of ensemble methods to select features. Authors studied different feature selection algorithms in their simple and ensemble implementation. They also investigated the effects of a data perturbation ensemble strategy.

Algorithms at this step often tries to decrease computations complexity by parallelizing the features selection process. However, many other problems should not be ignored, namely: when the number of features extremely exceeds the number of samples, it leads often to over fitting. other issue is the difficulty to handle imbalanced data, when there is big difference in the number of elements in each class.

3.1.4 Data mining

The data mining algorithm should be selected with respect to the goal of the KDD which was specified in the selection step. Other parameters and derives functions of this algorithm should be identified according to the current dataset. It is not always a trivial task to select the good algorithm for the data. But, defining the task helps in selecting the required algorithms. There are six main tasks of data mining: Classification, Clustering, Association, Summarization, and Prediction.

Despite the availability of big data infrastructures with the most popular machine learning algorithms, and the efficiency of the MapReduce paradigms to parallelize data processing, scientists still encounter many other challenges; on the one hand, MapReduce methodology cannot be directly applied to all algorithms. On the other hand, MapReduce considers that the mapped data is independent, and can be processed in parallel; which is not always true because some algorithms need to make interdependent predictions among data to take the final decision.

Researchers have been conducted to develop their specific algorithms, and refine existent ones to make them work in a big data context.

Authors in [63] proposed an hybrid algorithm, named PAK, in order to reduce the complexity of analyzing dimensional XML files. Instead of directly clustering a large document dataset, PAK selects only frequent documents using parallel Apriori algorithm, and then, applies k-means on the selected documents. The algorithm uses Euclidian distance to measure the similarity between frequent documents, and Dunn index to find the best number of clusters. based on Hadoop technology. The main drawback of PAK algorithm is the expensive computational cost of frequent pattern mining.

[64] provides an overview of scalable tensors mining algorithms, and their advantages. Most of multidimensional data can be modeled as arrays, however, the lack of scalable algo-

rithms and the difficulty of setting algorithms parameters are still challenging the use of these techniques.

Mining microarray data sets is considered as a big data challenge, since they have huge number of features that needs to be analyzed in real time. [65] presents a method for mining micro array datasets, by using statistical tests based on MapReduce to select relevant features, then it applies a MapReduce based K-nearest neighbor (mrKNN) on the selected features to classify the data into cancerous/non-cancerous samples. A major drawback of this method is the use of Hadoop's MapReduce implementation, which is considered very slow in comparison other MapReduce implementations.

In order to effectively extract knowledge from electronic health records, and facilitate the application of data mining techniques on it, authors in [66] discussed the need of data schema standardization. And proposed an architecture for preprocessing and transforming electronic health records to put them in a common portable format. This method addresses both volume and variety of data, however, standardization is not always a trivial task.

In [67] authors proposed WEPS algorithm, Weighted Erasable Pattern mining algorithm on Sliding window-based data streams. This algorithm can process dynamic data streams. It is based on sliding window and can find weights erasable patterns. It provides advantages in runtime, memory, pattern generation, and scalability.

[68] proposed an algorithm for dealing with big data, that is based on k-means and kNN methods. First, it uses k-means as a preliminary step to separate the dataset into multiple parts. And then it applies kNN on each part. This method has a linear complexity to the sample size. Its performances have been compared with traditional methods using classification accuracy and execution time.

Based on MapReduce, author in [69] designed a parallel implementation of the back-propagation neural network algorithm. This method uses particle swarm optimization algorithm (PSO) to optimize the neural network's initial weights and thresholds. The metrics used to measure the algorithm performances are the classification time and accuracy, applied on image dataset.

In order to decrease the time of processing a huge number of documents, [70] presents a new distributed architecture for scaling up text analysis by distributing algorithms over several virtual machines. Apache Storm framework is used to manage the modules inside virtual machines.

Based on decision trees and kNN algorithm, [71] presents and adaptive rule-based classifier (ARB) for classifying multi-class biological data. ARB deals with classification problems such as overfitting, noisy instances and class-imbalance data. Decisions trees were used to classify biological data, while kNN is used to detect the misclassified instances. The algorithm was implemented in Java using Weka [72], and authors used f-score measure to obtain the algorithm accuracy.

[73] proposed a text clustering method FC-DM. This algorithm performs a set of divide-and-merge operations on clusters, until it finds the adequate number of clusters. In this method, extended document features have been used, such as synonyms and co-occurring words. FC-DM Algorithm has been used for clustering news articles, and its accuracy measured using F-score.

These algorithms have shown good results in comparison to centralized and/or classical

methods which in some cases cannot be applicable. However, they still have problems in some practical applications. Machine learning algorithms can participate in solving various big data problems, however, most of them are operating in solving particular cases, and various enhancements should be done to make them applicable to a wide range of applications.
Table 3 summarizes the findings of this study.

3.1.5 Interpretation/Evaluation

The interpretation or evaluation, reveals weather the detected pattern is interesting and contains knowledge or not. In this last case, the cause has to be found out, by fall back to previous steps and trying other techniques.

The traditional known KM measurement mechanisms are: accuracy, recall, precision and f-measure. Big Data have its own specificity, we have other parameters that are more significant and should not be ignored like: system response time, scalability, consistency ,and the accessibility. Also the similarity measure techniques change with changing the context of big data, and varie upon the nature of the problem:

- Volume and time complexity: Metrics used here concern the dataset volume and processing time complexity. An efficient algorithm can handle large data in less computational time.
- Velocity: Check whether or not the algorithm can handle streaming data and take real time decisions.
- Variety: The type of dataset is an important criterion. Most algorithms are designed to process numerical data. However, real world datasets often contain also unstructured data that should not be ignored.
- Quality: Another metric that is used to evaluate big data algorithms is by evaluating their clustering quality. Which can be achieved by using methods like:
  - Statistical tests, such as ANOVA. This techniques are used to compare multiple means of groups and determine weather there exists any significant difference between them. ANOVA is a parametric method, in other words it makes assumptions about the groups (each sample is normally distributed, samples are drawn independently, with common variance,...etc.).

    Kruskal-Wallis is a test that makes no such assumptions. It can be considered as the non parametric alternative of ANOVA.
  - Cluster validity indices : They measure the similarity between two clustering algorithms. This techniques are used to compare how well different clustering algorithms perform on a set of data. As example of this methods : Dunn's index [74].
  - Classifier performances : Multiple measures can be extracted from the classification confusion matrix.
    * Recall : the proportion of positive cases that were correctly identified.
    * Accuracy: the proportion of the total number of predictions that were correct.

- ∗ The false positive rate (FPR): the proportion of negatives cases that were incorrectly classified as positive.
- ∗ True negative rate (TNR): the proportion of negatives cases that were classified correctly
- ∗ False negative rate (FNR): proportion of positives cases that were incorrectly classified as negative.
- ∗ Precision: proportion of the predicted positive cases that were correct
- ∗ Receiver Operating Characteristic (ROC) Curve: is a graphical approach for displaying the tradeoff between true positive rate (TPR) and false positive rate (FPR) of a classifier.

3.2 Data Intelligence in the context of big data

According to (B.Fernandez, et al 2014) Knowledge has been classified and characterized in several different ways :

- Procedural or Declarative Knowledge. The first distinction is that between declarative knowledge (facts) and procedural knowledge (how to ride a bicycle). Declarative knowledge (or substantive knowledge, as it is also called) focuses on beliefs about relationships among variables. Procedural knowledge, in contrast, focuses on beliefs relating sequences of steps or actions to desired (or undesired) outcomes
- Another important classification of knowledge views it as Tacit or Explicit . Explicit knowledge typically refers to knowledge that has been expressed into words and numbers. Such knowledge can be shared formally and systematically in the form of data, specifications, manuals, drawings, audio and videotapes, computer programs, patents, and the like. Tacit knowledge includes insights, intuitions, and hunches. It is difficult to express and formalize, and therefore difficult to share. Tacit knowledge is more likely to be personal and based on individual experiences and activities.

Combining the Classifications of Knowledge the above classifications of knowledge are independent. In other words, procedural knowledge could be either tacit or explicit and either general or specific. Similarly, declarative knowledge could be either tacit or explicit and either general or specific.

Making sense of large amounts of disorganized information, which is spread across wide swaths of an organization, has always been the defining challenge of knowledge management. First of all, knowledge management deals with each level of the pyramid presented in figure 2. Extracting useful knowledge from big data is the most powerful challenge, against others like data pre-processing, and representation etc.

The primary problem noted by [16] is how to set a link between big data and knowledge, they defined knowledge maturity, which describes how knowledge can be controlled and how the knowledge maturity model can serves as a platform to integrate knowledge with new product development in big data times. Many researchers [76], worked on the literature concerning knowledge management (KM) and intellectual capital (IC) to develop a vision of big data that fits with existing theory. Also they deal with every issue coming with big data like the difficulties to transform the huge quantity of non-structured information into generic knowledge [77], or structured data [78]. The capitalization of knowledge is dependent of the human context [79]. Authors in [104] discuss the same problem by proposing a way to overcome two fundamental issues: data heterogeneity and advanced processing capabilities. They presented a knowledge-based solution for Big Data analytics, which consists in applying automatic schema mapping to face data heterogeneity issues, as well as ontology extraction and semantic inference to support innovative processing. To do so, they designed and implemented a flexible architectural platform providing distributed mining solution for huge amounts of unstructured data within the context of complex event processing systems, allowing the easy integration of a large number of information sources geographically scattered throughout the world. The main idea in their work is designing a knowledge-based enforcement to publish/subscribe services in order to address their limitations in supporting syntactic and semantic interoperability among heterogeneous entities. The information stored from last experiences is very important. Based on these information, [80] presents a time series data mining methodology for temporal knowledge discovery in big BAS data; also they develop two methods for efficient post-processing of discovered knowledge. In [81], authors propose a framework allowing managing and generating knowledge from information on past experiences. They suggest an original Experience Feedback process dedicated to maintenance, allowing to capitalize on past activities by (i) formalizing the domain knowledge and experiences using a visual knowledge representation formalism with logical foundation (Conceptual Graphs); (ii) extracting new knowledge, and (iii) interpreting and evaluating it.

[82] Proposes a big data marching pattern, from the knowledge discovery view. [83] Suggested using a trace based system whose goal is to extract new knowledge rules about transitions and activities in the maintenance process.

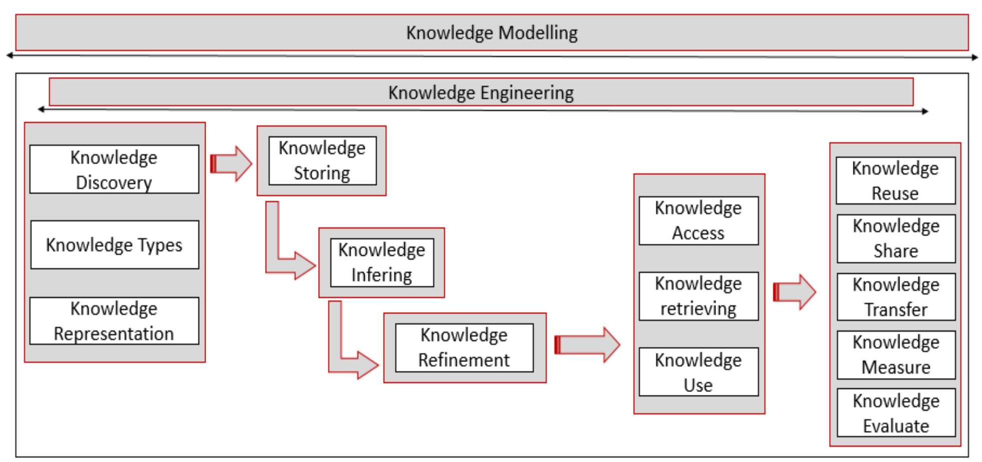

Fig. 4. Knowledge Management Process in big data context.

There are many types of knowledge such as temporal, textual, graphical, and semantic. Each kind presents a challenge especially in big data context due to different Vs, indeed they still have the main attention, for example, temporal discovered knowledge allows to identify dynamics, patterns and anomalies in building operations, derive temporal association rules within and between subsystems, assess building system performance and spot opportunities in energy conservation [6], also ontology extraction and semantic inference support innovative processing, and its usefulness responds to many issues in knowledge context. [7].

To survey the different knowledge management techniques in Big Data, we have used many keywords like knowledge management in big data context, knowledge management issues in big data, big data and knowledge management a survey, etc. The study results are shown on Table 4, as we remark according to the KM process; by projecting different works on KM features; we first start with knowledge modelling that got the higher attention of works, followed by knowledge types. As we explained previously, knowledge has different types, and each type has its specificity, and needs a specific way to deal with; then we'll find knowledge representation, which depends on knowledge type. Finally, the knowledge discovery has nearly the same importance of the previous two types. Those are the critical steps on knowledge management process; and in a parallel way we should think about three in the same time and in the same part of any architecture in big data context. Projecting the same works on big data axes (5Vs), we observe that: (i) the first V: volume is present on all works, (ii) those who deal with the velocity deal also with variety except three cases, (iii) for veracity, it's also the same case. The three first (Vs) take most attention because they define big data applications, and they are the critical problems to handle for extracting the value (fifth V); and to respond to the different issues mentioned earlier. Figure 5, presents the summarized study results by proportions; first the knowledge modelling attracts the most attention by 12% from the hole woks, 12% means the percentage of papers that discuss the knowledge modeling from the chosen papers, followed by knowledge types, and sharing by 11%. The main goal of this study is to better master the knowledge process in big data context. To summarize, we start by knowledge modeling coupled by knowledge representation; then, we discover knowledge. In the next step, we deal with the storage and the other aspects.

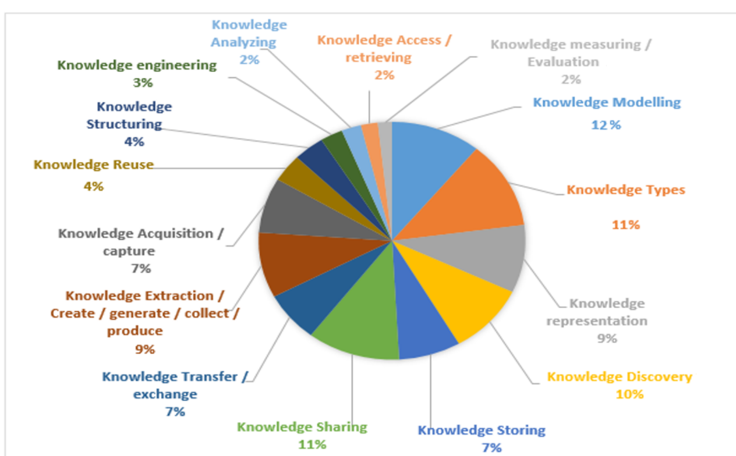

Fig. 5. Statistics for Knowledge management mechanisms

4 CONCLUSION

Data have been increasingly accumulated by different organizations in many forms. Analyzing this data will help making correct decisions and form actionable insights from data. However, mining big data is not always trivial to available algorithms and technology. In this paper, we have given a survey on different techniques used to extract knowledge out of high dimensional data, from the discovery phase until the knowledge processing. We have discussed big data characteristics and problems by investigating recent works in this field. We also explored infrastructures and algorithms that were developed to overcome big data complexity. We have proposed a framework for extracting knowledge from data. Although the achievements of big data analytics are admirable, we revealed through this study that big data mining is a challenging and emerging field. The principal issues with existing data mining algorithms concern: the lack of a scalable form, high complexity and computational costs, and the difficulty to apply it on other problems. In fact, this field has many unanswered questions, and still remains to be unlocked.

Table 3. A comparative table of data mining algorithms in big data context

Reference	Application Domain	Algorithms	Advantages	Inconvenients	Big Data addressed problems			Dataset	Technologies	Evaluation metrics
					Volume	Variety	Velocity			
[56] MR-EFS (2015)	General	EFS - CHC	Parallel Features Selecting	-	Y	-	-	Epsilon dataset by LIBSVM, & ECBDL14	Apache Spark	Area Under the Curve (AUC) & Training runtime
[57] HDBPSO (2014)	Gene expression data	Binary PSO using Hamming distance	Reduces time complexity	-	Y	-	-	colon cancer, defused B-cell lymphoma and leukemia	C Language + Weka	Classification Accuracy
[63] PAK (2015)	Web data mining	Apriori	Mining Big XML files	Memory & computational cost can still be very expensive	Y	Y	-	Wikipedia dataset	-	Dunn's Index
[64] (2015)	General	-	Mining multi-dimensional arrays	lack of scalable algorithms	Y	-	-	-	-	-
[65] (2016)	Microarray gene expression	Statistic tests+ kNN	Uses kNN MapReduce implementation	Uses Hadoop MapReduce implementation	Y	-	-	From NCBI GEO : GSE13159, GSE13204, GSE15061	Hadoop	Training accuracies using ANOVA, Kruskal-Wallis, and Friedman tests
[66] (2016)	Healthcare	-	Structuring data	Not always trivial	y	y	-	Collected (patient's information)	-	Consumed time and memory

(*Table 3. Continued*).

[67] WEPS (2016)	General	-	Allow algorithm to consider the latest information on a given data stream	May be difficult to provide real-time mining in case of large transactions	-	-	Y	15 different datasets	C++	runtime and memory usage on changing another metric
[68] (2015)	Medical imaging data	Kmeans then kNN	Scale kNN algorithm	-	Y	-	-	medical imaging datasets	-	Classification accuracy and Time cost
[69] (2016)	Imaging data	ANN & PSO	Parallel design of the algorithm	-	Y	-	-	SUN Database images	Hadoop	Classification accuracy & system efficiency
[75] DFS (2015)	microarray data	Multiple feature selection algorithms	Parallel Features Selecting	-	Y	-	-	Colon, DLBCL, CNS, Leukemia, Prostate, Lung, Ovarian, Breast	-	Accuracy & Time complexity
[60] ROSEFW-RF (2015)	Bio-informatics	Random Forest	Cope with imbalanced class distribution	-	Y	-	-	ECBDL'14	Hadoop	TPR * TNR
[70] (2014)	Natural Language Processing	Text mining	Distributed architecture for scaling up text analysis	Produces too much data traffic between modules	Y	Y	-	Car & Wikinews datasets	Storm	Processing time
[71] ARB (2016)	Genomic data	DT, kNN	deals with over-fitting, noisy instances and class-imbalance data	No parallel implementation	Y	-	-	Genomic datasets	Java & Weka	Classification accuracy using F-score
[73] FC-DM (2015)	Textual : news articles	K-means	Reduce dimensions by determining the number of clusters	Not parallelized	Y	-	-	provided by Sougou Lab	-	F-score

(-) refers to a lack of definition in the referenced papers.

Table 4. PROJECTION OF DIFFERENT WORKS ON (5V) BIG DATA FEATURES AND KM FEATURES.

REFERENCE	**Knowledge management features**															**Big Data features**				
	Modeling	**Types**	**Representation**	**Discovery**	**Storing**	**Sharing**	**Transfer**	**Extraction**	**Acquisition**	**Reuse**	**Structuring**	**Engineering**	**Analyzing**	**Access**	**Measuring**	**Volume**	**Variety**	**Velocity**	**Veracity**	**Value**
[84]	X		X	X				X					X			X	X	X	X	
[85]					X			X								X	X			
[86]							X									X	X	X		
[87]			X													X		X		
[88]				X												X	X	X	X	
[89]	X							X	X							X				X
[90]		X		X	X			X								X	X	X		X
[91]	X															X	X			X
[92]	X															X	X	X		X
[93]	X		X													X	X	X		
[94]				X				X								X	X	X		
[95]	X					X	X		X	X						X	X	X	X	X
[96]	X					X	X		X	X						X	X	X		X
[97]		X		X											X	X	X			
[98]		X		X	X											X	X	X		
[99]	X															X	X	X		
[100]	X			X												X	X	X	X	X
[101]	X															X	X	X	X	X
[102]	X	X				X		X	X	X	X	X		X		X	X	X		
[103]		X	X					X									X	X		
[104]		X		X	X											X	X	X		
[105]	X															X	X			X
[106]				X												X	X			X
[107]				X											X	X	X			X
[108]				X												X	X	X		
[109]	X												X			X	X	X		X

References

1. J. H. Friedman (1998), *Data mining and statistics: What's the connection?*, Computing Science and Statistics, vol. 29, pp. 3–9.
2. K. W. Lin and Y.-C. Lo (2013), *Efficient algorithms for frequent pattern mining in many-task computing environments*, Knowledge-Based Systems, vol. 49, pp. 10–21.
3. D. Ioannidis, P. Tropios, S. Krinidis, G. Stavropoulos, D. Tzovaras and S. Likothanasis (2016), *Occupancy driven building performance assessment*, Journal of Innovation in Digital Ecosystems, vol. 3, pp. 57–69.
4. A. Loureiro, L. Torgo and C. Soares (2004), *Outlier detection using clustering methods: a data cleaning application*, in *Proceedings of KDNet Symposium on Knowledge-based systems for the Public Sector*.
5. M. Kirlidog and C. Asuk (2012), *A fraud detection approach with data mining in health insurance*, Procedia-Social and Behavioral Sciences, vol. 62, pp. 989–994.
6. M. N. Mohammad, N. Sulaiman and O. A. Muhsin (2011), *A novel intrusion detection system by using intelligent data mining in weka environment*, Procedia Computer Science, vol. 3, pp. 1237–1242.
7. S. Duque and M. N. bin Omar (2015), *Using data mining algorithms for developing a model for intrusion detection system (IDS)*, Procedia Computer Science, vol. 61, pp. 46–51.
8. R. Agrawal, T. Imieliński and A. Swami (1993), *Mining association rules between sets of items in large databases*, in *Acm sigmod record*, vol. 22, pp. 207–216, ACM.
9. J. Soni, U. Ansari, D. Sharma and S. Soni (2011), *Predictive data mining for medical diagnosis: An overview of heart disease prediction*, International Journal of Computer Applications, vol. 17, pp. 43–48.
10. S. R. D. C. T. S. Doddi, Achla Marathe (2001), *Discovery of association rules in medical data*, Medical informatics and the Internet in medicine, vol. 26, pp. 25–33.
11. N. Gupta, N. Mangal, K. Tiwari and P. Mitra (2006), *Mining quantitative association rules in protein sequences*, in *Data Mining*, pp. 273–281, Springer.
12. T. Oyama, K. Kitano, K. Satou and T. Ito (2002), *Extraction of knowledge on protein–protein interaction by association rule discovery*, Bioinformatics, vol. 18, pp. 705–714.
13. J. Deshmukh and U. Bhosle (2016), *Image Mining Using Association Rule for Medical Image Dataset*, Procedia Computer Science, vol. 85, pp. 117–124.
14. J. A. Rushing, H. Ranganath, T. H. Hinke and S. J. Graves (2002), *Image segmentation using association rule features*, IEEE Transactions on Image Processing, vol. 11, pp. 558–567.
15. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth et al. (1996), *Knowledge Discovery and Data Mining: Towards a Unifying Framework.*, in *KDD*, vol. 96, pp. 82–88.
16. I. Becerra-Fernandez and R. Sabherwal (2014), *Knowledge management: Systems and processes*, Routledge.
17. Q. Yang, X. Wu, P. Domingos, C. Elkan, J. Gehrke, J. Han, D. Heckerman, D. Keim, J. Liu, D. Madigan, G. Piatetsky-Shapiro, V. V. Raghavan, R. Rastogi, S. J. Stolfo, A. Tuzhilin and B. W. Wah (2006), *10 Challenging Problems in Data Mining Research*, International Journal of Information Technology & Decision Making, vol. 5, pp. 597–604, ISSN 0219-6220, doi: 10.1142/S0219622006002258.
18. A. Cuzzocrea, I.-Y. Song and K. C. Davis (2011), *Analytics over large-scale multidimensional data: the big data revolution!*, … 14th international workshop on Data …, pp. 101–104, doi: 10.1145/2064676.2064695.
19. G. G.-h. Lin and J. G. Scott (2012), *NIH Public Access*, vol. 100, pp. 130–134, ISSN 15378276, doi:10.1016/j.pestbp.2011.02.012.Investigations.
20. Q. He, N. Li, W. J. Luo and Z. Z. Shi (2014), *A survey of machine learning algorithms for big data*, Moshi Shibie yu Rengong Zhineng/Pattern Recognition and Artificial Intelligence, vol. 27, pp. 327–336, ISSN 10036059, doi:10.1186/s13634-016-0355-x.
21. A. Jha (2016), *A Review on the Study and Analysis of Big Data using Data Mining Techniques*,

vol. 7.

22. U. Sivarajah, M. M. Kamal, Z. Irani and V. Weerakkody (2016), *Critical analysis of Big Data challenges and analytical methods*, Journal of Business Research, ISSN 01482963, doi: 10.1016/j.jbusres.2016.08.001.
23. S. Aridhi and E. Mephu Nguifo (2016), *Big Graph Mining: Frameworks and Techniques*, Big Data Research, vol. 1, pp. 1–10, ISSN 22145796, doi:10.1016/j.bdr.2016.07.002.
24. I. Yaqoob, I. A. T. Hashem, A. Gani, S. Mokhtar, E. Ahmed, N. B. Anuar and A. V. Vasilakos (2016), *Big data: From beginning to future*, International Journal of Information Management, vol. 36, pp. 1231–1247, ISSN 02684012, doi:10.1016/j.ijinfomgt.2016.07.009.
25. H. Chen, R. H. L. Chiang and V. C. Storey (2012), *Business Intelligence and Analytics: From Big Data to Big Impact*, vol. 36.
26. M. Injadat, F. Salo and A. B. Nassif (2016), *Data Mining Techniques in Social Media: A Survey Data Mining Techniques in Social Media: A Survey*, Neurocomputing, pp. 1–17, ISSN 09252312, doi:10.1016/j.neucom.2016.06.045.
27. K. Sin and L. Muthu (2015), *Application of big data in education data mining and learning analytics-A literature review*, Ictact Journal on Soft Computing: Special Issue on Soft Computing Models for Big Data, vol. 5, pp. 1035–1049.
28. J. T. Wassan (2015), *Discovering Big Data Modelling for Educational World*, Procedia - Social and Behavioral Sciences, vol. 176, pp. 642–649, ISSN 1877-0428, doi: http://dx.doi.org/10.1016/j.sbspro.2015.01.522.
29. R. Addo-Tenkorang and P. T. Helo (2016), *Big Data Applications in Operations/Supply-Chain Management: A Literature Review*, Computers & Industrial Engineering, pp. –, ISSN 0360-8352, doi:http://dx.doi.org/10.1016/j.cie.2016.09.023.
30. M. Bilal, L. O. Oyedele, J. Qadir, K. Munir, S. O. Ajayi, O. O. Akinade, H. A. Owolabi, H. A. Alaka and M. Pasha (2016), *Big Data in the construction industry: A review of present status, opportunities, and future trends*, Advanced Engineering Informatics, vol. 30, pp. 500–521, ISSN 14740346, doi:10.1016/j.aei.2016.07.001.
31. I. de la Torre D??ez, H. M. Cosgaya, B. Garcia-Zapirain and M. L??pez-Coronado (2016), *Big Data in Health: a Literature Review from the Year 2005*, Journal of Medical Systems, vol. 40, pp. 1–6, ISSN 1573689X, doi:10.1007/s10916-016-0565-7.
32. S. Fodeh and Q. Zeng (2016), *Mining Big Data in biomedicine and health care*, Journal of Biomedical Informatics, vol. 63, pp. 400–403, ISSN 15320464, doi:10.1016/j.jbi.2016.09.014.
33. A. Elragal and M. Haddara (2014), *Big Data Analytics : a Text Mining-Based Literature Analysis*, Norsk konferanse for organisasjoners bruk av IT, vol. 22, pp. 1–12.
34. S. Goswami, S. Chakraborty, S. Ghosh, A. Chakrabarti and B. Chakraborty (2016), *A review on application of data mining techniques to combat natural disasters*, Ain Shams Engineering Journal, ISSN 20904479, doi:10.1016/j.asej.2016.01.012.
35. A. Gandomi and M. Haider (2015), *Beyond the hype: Big data concepts, methods, and analytics*, International Journal of Information Management, vol. 35, pp. 137–144.
36. T. Huang, L. Lan, X. Fang, P. An, J. Min and F. Wang (2015), *Promises and challenges of big data computing in health sciences*, Big Data Research, vol. 2, pp. 2–11.
37. K. Normandeau (2013), *Beyond volume, variety and velocity is the issue of big data veracity*, Inside Big Data.
38. M. Götz, M. Richerzhagen, C. Bodenstein, G. Cavallaro, P. Glock, M. Riedel and J. A. Benediktsson (2015), *On scalable data mining techniques for earth science*, Procedia Computer Science, vol. 51, pp. 2188–2197, ISSN 18770509, doi:10.1016/j.procs.2015.05.494.
39. J. Dean and S. Ghemawat (2008), *MapReduce*, Communications of the ACM, vol. 51, p. 107, ISSN 00010782, doi:10.1145/1327452.1327492.
40. C.-F. Tsai, W.-C. Lin and S.-W. Ke (2016), *Big Data Mining with Parallel Computing: A Comparison of Distributed and MapReduce Methodologies*, Journal of Systems and Software, vol. 122, pp. 83–92, ISSN 01641212, doi:10.1016/j.jss.2016.09.007.
41. Yahoo (2014), *Apache Hadoop*.

42. M. Zaharia, M. Chowdhury, T. Das and A. Dave (2012), *Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing*, in *NSDI'12 Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pp. 2–2, USENIX Association, ISBN 978-931971-92-8, ISSN 00221112, doi:10.1111/j.1095-8649.2005.00662.x.
43. The Apache Software Foundation (2015), *Apache Storm*, URL `http://storm.apache.org/`.
44. G. D. F. Morales and A. Bifet (2015), *SAMOA: Scalable Advanced Massive Online Analysis*, Journal of Machine Learning Research, vol. 16, pp. 149–153, ISSN 15337928.
45. Apache Software Foundation (2015), *Apache Flink*, URL `http://flink.apache.org/`.
46. A. Alexandrov, R. Bergmann, S. Ewen, J. C. Freytag, F. Hueske, A. Heise, O. Kao, M. Leich, U. Leser, V. Markl, F. Naumann, M. Peters, A. Rheinländer, M. J. Sax, S. Schelter, M. Höger, K. Tzoumas and D. Warneke (2014), *The Stratosphere platform for big data analytics*, VLDB Journal, vol. 23, pp. 939–964, ISSN 0949877X, doi:10.1007/s00778-014-0357-y.
47. *Apache h2o*, URL `http://www.h2o.ai/`.
48. S. Ghemawat, H. Gobioff and S.-T. Leung (2003), *The Google file system*, ACM SIGOPS Operating Systems Review, vol. 37, p. 29, ISSN 01635980, doi:10.1145/1165389.945450.
49. K. Shvachko, H. Kuang, S. Radia and R. Chansler (2010), *The Hadoop distributed file system*, 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies, MSST2010, pp. 1–10, ISSN 978-1-4244-7152-2, doi:10.1109/MSST.2010.5496972.
50. A. Makris, K. Tserpes, V. Andronikou and D. Anagnostopoulos (2016), *A Classification of NoSQL Data Stores Based on Key Design Characteristics*, Procedia Computer Science, vol. 97, pp. 94–103.
51. A. Corbellini, C. Mateos, A. Zunino, D. Godoy and S. Schiaffino (2017), *Persisting big-data: The NoSQL landscape*, Information Systems, vol. 63, pp. 1–23.
52. I. Guyon and A. Elisseeff (2003), *An introduction to variable and feature selection*, Journal of machine learning research, vol. 3, pp. 1157–1182.
53. H. Liu and H. Motoda (2012), *Feature selection for knowledge discovery and data mining*, vol. 454, Springer Science & Business Media.
54. L. Wang, Y. Wang and Q. Chang (2016), *Feature selection methods for big data bioinformatics: A survey from the search perspective*, Methods, vol. 111, pp. 21–31.
55. V. Bolón-Canedo, N. Sánchez-Maroño and A. Alonso-Betanzos (2015), *Recent advances and emerging challenges of feature selection in the context of big data*, Knowledge-Based Systems, vol. 86, pp. 33–45.
56. D. Peralta, S. Del Río, S. Ramírez-Gallego, I. Triguero, J. M. Benitez and F. Herrera (2015), *Evolutionary Feature Selection for Big Data Classification: A MapReduce Approach*, Mathematical Problems in Engineering, vol. 2015, pp. 1–11, ISSN 15635147, doi:10.1155/2015/246139.
57. H. Banka and S. Dara (2015), *A Hamming distance based binary particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation*, Pattern Recognition Letters, vol. 52, p. 94100, ISSN 01678655, doi:10.1016/j.patrec.2014.10.007.
58. V. Bolon-Canedo, N. Sanchez-Marono and A. Alonso-Betanzos (2015), *Distributed feature selection: An application to microarray data classification*, Applied Soft Computing, vol. 30, p. 136150, ISSN 15684946, doi:10.1016/j.asoc.2015.01.035.
59. M. Moradkhani, A. Amiri, M. Javaherian and H. Safari (2015), *A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSSr algorithm*, Applied Soft Computing, vol. 35, p. 123135, ISSN 15684946, doi:10.1016/j.asoc.2015.03.049.
60. I. Triguero, S. del Río, V. López, J. Bacardit, J. M. Benítez and F. Herrera (2015), *ROSEFW-RF: The winner algorithm for the ECBDL'14 big data competition: An extremely imbalanced big data bioinformatics problem*, Knowledge-Based Systems, vol. 87, pp. 69–79, ISSN 09507051, doi:10.1016/j.knosys.2015.05.027.
61. J. Apolloni, G. Leguizamn and E. Alba (2016), *Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments*, Applied Soft Computing, vol. 38, p. 922932, ISSN 15684946, doi:10.1016/j.asoc.2015.10.037.
62. B. Pes, N. Dess and M. Angioni (2017), *Exploiting the ensemble paradigm for stable feature selection: A case study on high-dimensional genomic data*, Information Fusion, vol. 35, p. 132147,

ISSN 15662535, doi:10.1016/j.inffus.2016.10.001.

63. A. Muralidhar and V. Pattabiraman (2015), *An efficient association rule based clustering of XML documents*, Procedia Computer Science, vol. 50, pp. 401–407, ISSN 18770509, doi: 10.1016/j.procs.2015.04.024.
64. L. Sael, I. Jeon and U. Kang (2015), *Scalable Tensor Mining*, Big Data Research, vol. 2, pp. 82–86, ISSN 22145796, doi:10.1016/j.bdr.2015.01.004.
65. M. Kumar, N. K. Rath and S. K. Rath (2016), *Analysis of microarray leukemia data using an efficient MapReduce-based K-nearest-neighbor classifier*, Journal of Biomedical Informatics, vol. 60, pp. 395–409, ISSN 15320464, doi:10.1016/j.jbi.2016.03.002.
66. S. Batra and S. Sachdeva (2016), *Organizing standardized electronic healthcare records data for mining*, Health Policy and Technology, pp. 1–17, ISSN 22118845, doi:10.1016/j.hlpt.2016.03.006.
67. U. Yun and G. Lee (2016), *Sliding window based weighted erasable stream pattern mining for stream data applications*, Future Generation Computer Systems, vol. 59, pp. 1–20, ISSN 0167739X, doi: 10.1016/j.future.2015.12.012.
68. Z. Deng, X. Zhu, D. Cheng, M. Zong and S. Zhang (2015), *Efficient kNN classification algorithm for big data*, Neurocomputing, ISSN 18728286, doi:10.1016/j.neucom.2015.08.112.
69. J. Cao, H. Cui, H. Shi and L. Jiao (2016), *Big data: A parallel particle swarm optimization-back-propagation neural network algorithm based on MapReduce*, PLoS ONE, vol. 11, pp. 1–17, ISSN 19326203, doi:10.1371/journal.pone.0157551.
70. R. Agerri, X. Artola, Z. Beloki, G. Rigau and A. Soroa (2014), *Big data for Natural Language Processing: A streaming approach*, Knowledge-Based Systems, vol. 79, pp. 36–42, ISSN 09507051, doi:10.1016/j.knosys.2014.11.007.
71. D. M. Farid, M. A. Al-Mamun, B. Manderick and A. Nowe (2016), *An adaptive rule-based classifier for mining big biological data*, Expert Systems with Applications, vol. 64, pp. 305–316, ISSN 09574174, doi:10.1016/j.eswa.2016.08.008.
72. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten (2009), *The WEKA data mining software*, ACM SIGKDD Explorations, vol. 11, pp. 10–18, ISSN 19310145, doi:10.1145/1656274.1656278.
73. M. Yuan and Y. Shi (2015), *Text clustering based on a divide and merge strategy*, Procedia Computer Science, vol. 55, pp. 825–832, ISSN 18770509, doi:10.1016/j.procs.2015.07.153.
74. D. L. Davies and D. W. Bouldin (1979), *A cluster separation measure*, IEEE transactions on pattern analysis and machine intelligence, pp. 224–227.
75. V. Bol??n-Canedo, N. S??nchez-Maro??o and A. Alonso-Betanzos (2015), *Distributed feature selection: An application to microarray data classification*, Applied Soft Computing Journal, vol. 30, pp. 136–150, ISSN 15684946, doi:10.1016/j.asoc.2015.01.035.
76. X. Yang (2015), *Knowledge management in big data times*, in *Big Data and Cloud Computing (BDCloud), 2015 IEEE Fifth International Conference on*, pp. 168–171, IEEE.
77. S. Tichkiewitch (2008), *Capitalization and reuse of forging knowledge in integrated design*, Methods and Tools for Effective Knowledge Life-Cycle-Management, pp. 479–485.
78. R. K. Lomotey and R. Deters (2014), *Towards knowledge discovery in big data*, in *Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on*, pp. 181–191, IEEE.
79. C. Esposito, M. Ficco, F. Palmieri and A. Castiglione (2015), *A knowledge-based platform for Big Data analytics based on publish/subscribe services and stream processing*, Knowledge-Based Systems, vol. 79, pp. 3–17.
80. P. P. Ruiz, B. K. Foguem and B. Grabot (2014), *Generating knowledge in maintenance from Experience Feedback*, Knowledge-Based Systems, vol. 68, pp. 4–20.
81. U. Shafique and H. Qaiser (2014), *A comparative study of data mining process models (KDD, CRISP-DM and SEMMA)*, Int. J. Innov. Sci. Res, vol. 12, pp. 217–222.
82. M.-H. Karray, B. Chebel-Morello and N. Zerhouni (2014), *PETRA: Process Evolution using a TRAce-based system on a maintenance platform*, Knowledge-Based Systems, vol. 68, pp. 21–39.
83. J. Dean and S. Ghemawat (2008), *MapReduce: simplified data processing on large clusters*, Communications of the ACM, vol. 51, pp. 107–113.

84. G. Bello-Orgaz, J. J. Jung and D. Camacho (2016), *Social big data: Recent achievements and new challenges*, Information Fusion, vol. 28, pp. 45–59.
85. R. Chalh, Z. Bakkoury, D. Ouazar and M. D. Hasnaoui (2015), *Big data open platform for water resources management*, in *Cloud Technologies and Applications (CloudTech), 2015 International Conference on*, pp. 1–8, IEEE.
86. J. Andreu-Perez, C. C. Poon, R. D. Merrifield, S. T. Wong and G.-Z. Yang (2015), *Big data for health*, IEEE journal of biomedical and health informatics, vol. 19, pp. 1193–1208.
87. K. Abuosba (2015), *Formalizing big data processing lifecycles: Acquisition, serialization, aggregation, analysis, mining, knowledge representation, and information dissemination*, in *Computing and Communication (IEMCON), 2015 International Conference and Workshop on*, pp. 1–4, IEEE.
88. G. Chatzigeorgakidis, S. Karagiorgou, S. Athanasiou and S. Skiadopoulos (2015), *A MapReduce based k-NN joins probabilistic classifier*, in *Big Data (Big Data), 2015 IEEE International Conference on*, pp. 952–957, IEEE.
89. I. Gorton, J. Klein and A. Nurgaliev (2015), *Architecture knowledge for evaluating scalable databases*, in *Software Architecture (WICSA), 2015 12th Working IEEE/IFIP Conference on*, pp. 95–104, IEEE.
90. M. El Houari, M. Rhanoui and B. El Asri (2015), *From Big Data to Big Knowledge: The art of making Big Data alive*, in *Cloud Technologies and Applications (CloudTech), 2015 International Conference on*, pp. 1–6, IEEE.
91. Y. Huang and X. Zhou (2015), *Knowledge model for electric power big data based on ontology and semantic web*, CSEE Journal of Power and Energy Systems, vol. 1, pp. 19–27.
92. M. D. Wang (2015), *Biomedical Big Data Analytics for Patient-Centric and Outcome-Driven Precision Health*, in *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, vol. 3, pp. 1–2, IEEE.
93. K. Taneja, Q. Zhu, D. Duggan and T. Tung (2015), *Linked enterprise data model and its use in real time analytics and context-driven data discovery*, in *Mobile Services (MS), 2015 IEEE International Conference on*, pp. 277–283, IEEE.
94. N. Bari, R. Vichr, K. Kowsari and S. Berkovich (2014), *23-bit metaknowledge template towards big data knowledge discovery and management*, in *Data Science and Advanced Analytics (DSAA), 2014 International Conference on*, pp. 519–526, IEEE.
95. C. L. Borgman, P. T. Darch, A. E. Sands, J. C. Wallis and S. Traweek (2014), *The ups and downs of knowledge infrastructures in science: Implications for data management*, in *Digital Libraries (JCDL), 2014 IEEE/ACM Joint Conference on*, pp. 257–266, IEEE.
96. M. A. Roger, Y. Xu and M. Zhao (2014), *BigCache for big-data systems*, in *Big Data (Big Data), 2014 IEEE International Conference on*, pp. 189–194, IEEE.
97. R. Y. Zhong, G. Q. Huang and Q. Dai (2014), *A big data cleansing approach for n-dimensional RFID-Cuboids*, in *Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the 2014 IEEE 18th International Conference on*, pp. 289–294, IEEE.
98. N. Mishra, C.-C. Lin and H.-T. Chang (2014), *A cognitive oriented framework for IoT big-data management prospective*, in *Communication Problem-Solving (ICCP), 2014 IEEE International Conference on*, pp. 124–127, IEEE.
99. A. Madkour, W. G. Aref and S. Basalamah (2013), *Knowledge cubesA proposal for scalable and semantically-guided management of Big Data*, in *Big Data, 2013 IEEE International Conference on*, pp. 1–7, IEEE.
100. C. Seebode, M. Ort, C. Regenbrecht and M. Peuker (2013), *BIG DATA infrastructures for pharmaceutical research*, in *Big Data, 2013 IEEE International Conference on*, pp. 59–63, IEEE.
101. P. Tin, T. T. Zin, T. Toriu and H. Hama (2013), *An Integrated Framework for Disaster Event Analysis in Big Data Environments*, in *Intelligent Information Hiding and Multimedia Signal Processing, 2013 Ninth International Conference on*, pp. 255–258, IEEE.
102. G. Stalidis and D. Karapistolis (2014), *Tourist destination marketing supported by electronic capitalization of knowledge*, Procedia-Social and Behavioral Sciences, vol. 148, pp. 110–118.
103. R. W. Gehl (2015), *Sharing, knowledge management and big data: A partial genealogy of the data*

scientist, European Journal of Cultural Studies, vol. 18, pp. 413–428.

104. C. Fan, F. Xiao, H. Madsen and D. Wang (2015), *Temporal knowledge discovery in big {BAS} data for building energy management*, Energy and Buildings, vol. 109, pp. 75 – 89, ISSN 0378-7788, doi:https://doi.org/10.1016/j.enbuild.2015.09.060, URL `http://www.sciencedirect.com/science/article/pii/S0378778815302991`.
105. X. Yang (2015), *Knowledge management in big data times.*
106. E. Begoli (2012), *A short survey on the state of the art in architectures and platforms for large scale data analysis and knowledge discovery from data*, in *Proceedings of the WICSA/ECSA 2012 Companion Volume*, pp. 177–183, ACM.
107. C. Fan, F. Xiao and C. Yan (2015), *A framework for knowledge discovery in massive building automation data and its application in building diagnostics*, Automation in Construction, vol. 50, pp. 81–90.
108. M. Castelli, L. Vanneschi, L. Manzoni and A. Popovič (2016), *Semantic genetic programming for fast and accurate data knowledge discovery*, Swarm and Evolutionary Computation, vol. 26, pp. 1–7.
109. A. Endert, S. Szymczak, D. Gunning and J. Gersh (2014), *Modeling in big data environments*, in *Proceedings of the 2014 Workshop on Human Centered Big Data Research*, p. 56, ACM.