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 level, 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
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>
<thead>
<tr>
<th>Table 1. A comparative study between big data analytics platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Current version</strong></td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td>Java, Python</td>
</tr>
<tr>
<td>Java, Scala, Python, R</td>
</tr>
<tr>
<td>Java, Scala, Python, Ruby, Clojure</td>
</tr>
<tr>
<td>Streaming support</td>
</tr>
</tbody>
</table>

### 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

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

#### 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 backpropagation 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 serve 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 capabili-
ties. 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 scat-
tered 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 se-
ries 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 experi-
ences 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.

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

![Fig. 4. Knowledge Management Process in big data context.](image-url)
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

<table>
<thead>
<tr>
<th>Reference</th>
<th>Application Domain</th>
<th>Algorithms</th>
<th>Advantages</th>
<th>Inconvenients</th>
<th>Big Data addressed problems</th>
<th>Dataset</th>
<th>Technologies</th>
<th>Evaluation metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>[63] PAK (2015)</td>
<td>Web data mining</td>
<td>Apriori Mining</td>
<td>Big XML files</td>
<td>Memory &amp; computational cost can still be very expensive</td>
<td>Y</td>
<td>Y</td>
<td>Wikipedia dataset</td>
<td>-</td>
</tr>
<tr>
<td>[64] (2015)</td>
<td>General</td>
<td>-</td>
<td>Mining multi-dimensional arrays</td>
<td>lack of scalable algorithms</td>
<td>Y</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[65] (2016)</td>
<td>Microarray gene expression</td>
<td>Statistic tests + kNN</td>
<td>Uses kNN MapReduce implementation</td>
<td>Uses Hadoop MapReduce implementation</td>
<td>Y</td>
<td>-</td>
<td>From NCBI GEO : GSE13159, GSE13204, GSE15061</td>
<td>Hadoop</td>
</tr>
<tr>
<td>[66] (2016)</td>
<td>Healthcare</td>
<td>-</td>
<td>Structuring data</td>
<td>Not always trivial</td>
<td>y</td>
<td>y</td>
<td>Collected (patient’s information)</td>
<td>-</td>
</tr>
</tbody>
</table>
### Table 3. Continued.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Domain/Technique</th>
<th>Data</th>
<th>Algorithm</th>
<th>Specific Information</th>
<th>C++</th>
<th>Runtime/Memory Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>[67] WEPS (2016)</td>
<td>General</td>
<td>-</td>
<td>Allow algorithm to consider the latest information on a given data stream</td>
<td>May be difficult to provide real-time mining in case of large transactions</td>
<td>-</td>
<td>Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference</th>
<th>Domain/Technique</th>
<th>Data</th>
<th>Algorithm/Technique</th>
<th>Specific Information</th>
<th>C++</th>
<th>Runtime/Memory Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>[68] (2015)</td>
<td>Medical imaging data</td>
<td>Kmeans then kNN</td>
<td>Scale kNN algorithm</td>
<td>-</td>
<td>Y</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference</th>
<th>Domain/Technique</th>
<th>Data</th>
<th>Algorithm/Technique</th>
<th>Specific Information</th>
<th>C++</th>
<th>Runtime/Memory Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>[69] (2016)</td>
<td>Imaging data</td>
<td>ANN &amp; PSO</td>
<td>Parallel design of the algorithm</td>
<td>-</td>
<td>Y</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference</th>
<th>Domain/Technique</th>
<th>Data</th>
<th>Algorithm/Technique</th>
<th>Specific Information</th>
<th>C++</th>
<th>Runtime/Memory Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>[75] DFS (2015)</td>
<td>microarray data</td>
<td>Multiple feature selection algorithms</td>
<td>Parallel Features Selecting</td>
<td>-</td>
<td>Y</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference</th>
<th>Domain/Technique</th>
<th>Data</th>
<th>Algorithm/Technique</th>
<th>Specific Information</th>
<th>C++</th>
<th>Runtime/Memory Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>[60] ROSEFW-RF (2015)</td>
<td>Bioinformatics</td>
<td>Random Forest</td>
<td>Cope with imbalanced class distribution</td>
<td>-</td>
<td>Y</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference</th>
<th>Domain/Technique</th>
<th>Data</th>
<th>Algorithm/Technique</th>
<th>Specific Information</th>
<th>C++</th>
<th>Runtime/Memory Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>[70] (2014)</td>
<td>Natural Language Processing</td>
<td>Text mining</td>
<td>Distributed architecture for scaling up text analysis</td>
<td>Produces too much data traffic between modules</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference</th>
<th>Domain/Technique</th>
<th>Data</th>
<th>Algorithm/Technique</th>
<th>Specific Information</th>
<th>C++</th>
<th>Runtime/Memory Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>[71] ARB (2016)</td>
<td>Genomic data</td>
<td>DT, kNN</td>
<td>deals with over-fitting, noisy instances and class imbalance data</td>
<td>No parallel implementation</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference</th>
<th>Domain/Technique</th>
<th>Data</th>
<th>Algorithm/Technique</th>
<th>Specific Information</th>
<th>C++</th>
<th>Runtime/Memory Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>[73] FC-DM (2015)</td>
<td>Textual : news articles</td>
<td>K-means</td>
<td>Reduce dimensions by determining the number of clusters</td>
<td>Not parallelized</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

(-) refers to a lack of definition in the referenced papers.
Table 4. PROJECTION OF DIFFERENT WORKS ON (5V) BIG DATA FEATURES AND KM FEATURES.

<table>
<thead>
<tr>
<th>REFERENCE</th>
<th>Knowledge management features</th>
<th>Big Data features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Modeling</td>
<td>Types</td>
</tr>
<tr>
<td>[84]</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>[85]</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>[86]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[87]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[88]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[89]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[90]</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>[91]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[92]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[93]</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>[94]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[95]</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>[96]</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>[97]</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>[98]</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>[99]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[100]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[101]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[102]</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>[103]</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>[104]</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>[105]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[106]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[107]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[108]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[109]</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
References


41. Yahoo (2014), *Apache Hadoop*. 

vol. 7.


47. Apache h2o, URL [http://www.h2o.ai/](http://www.h2o.ai/).


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.
103. R. W. Gehl (2015), *Sharing, knowledge management and big data: A partial genealogy of the data*


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.

