International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 3 (2015) pp. 5681-5705

http://www.ripublication.com

An Encyclopedic Overview of 'Big Data' *Analytics*

G. Bharadwaja Kumar

School of Computing Science & Engineering,
VIT University Chennai Campus, Chennai - 600127, India.
Email: bharadwaja.kumar@vit.ac.in

Abstract

In the recent years, the field of 'big data' has emerged as the new frontier in the wide spectrum of IT-enabled innovations and opportunities due to the information revolution. The ever-increasing creation of massive amounts of data through diversified data generating sources has prompted industry as well as academics to intensify their attention to harness and analyze big data. Big data technologies are maturing to a point in which more organizations are adapting big data as a core component of the information management and analytics infrastructure. Big data, as a compendium of emerging disruptive tools and technologies, emerged as the next big technological idea enabling integrated analytics in many common business scenarios. The objective of this paper is to give a comprehensive overview of challenges and opportunities of big data. This paper discuss in detail about various characteristics of big data and how to overcome the challenges posed by them. It also discusses various machine learning and data mining algorithms used to uncover hidden patterns in large datasets and how the big data analytics can help in various fields to take insightful decisions.

Keywords: Big data, 5 V's, Hadoop, MapReduce, Analytics.

1. Introduction

Big data is the recent buzzword in computer science research as well as in industry because of the exponential growth and availability of data. World Economic Forum declared data as a new class of economic asset, like currency or gold [1]. The digital/electronics revolution is not only causing enormous increase in volume (Exabytes to Zettabytes) of the data but also in variety (i.e. structured, semi structured and unstructured) and velocity (millions of devices and applications generating humongous data every second). This enormous growth in data is hitting the limits of existing information management infrastructures, forcing companies to invest in more

hardware and costly upgrades of databases and data warehouses. For many decades, people followed scale up approach that is using more and more high end servers and super computers for data intensive tasks. In many cases, adding traditional infrastructure is impractical because of high costs, scalability limitations when dealing with petabytes, and incompatibility of relational systems with unstructured data.

To tackle the enormous amount of data on web, Google came out with MapReduce programming model for parallelized distributed computing on server clusters. Google published the Google File System paper in October 2003 and the MapReduce paper in December 2004. Over the course of a few months, Doug Cutting and Cafarella built underlying file system and processing framework based on the inspiration given by Google research papers and ported Nutch (a search engine) on top of it. In 2006, when Doug Cutting was working with Yahoo, they spun out the storage and processing parts of Nutch to form Hadoop as an open-source Apache Software Foundation project. Since then Hadoop has become the de facto standard for storing, processing and analyzing hundreds of terabytes, and even petabytes of data.

Apache Hadoop is 100% open source, and pioneered a fundamentally new way of storing and processing data. Instead of relying on expensive, proprietary hardware, Hadoop enables distributed parallel processing of humongous data across a large number of inexpensive low-end servers that both store and process the data, and can scale without limits. This approach is called scale out approach. For example, Hadoop clusters at Yahoo! span over 40,000 servers, and store 40 petabytes of application data. After the release of Hadoop framework, 'Big Data Analytics' emerged as the next big technological idea [2] that enable businesses to take the strategic leap from hindsight to foresight by extracting value from the large datasets.

Many people say that 'big data' is big hype, but it is a fallacious notion. For decades, companies have been making business decisions based on transactional data stored in relational databases. However, there is a potential treasure-trove of non-traditional, unstructured data such as web logs, social media, email, sensor data, image, audio and video that can be mined for useful information. On the other hand, data-driven decision making has grown exponentially in recent years and gut instinct has become less of a driving factor in all industries. Additionally, the 'Smart City' concept, a confluence of the 'Internet of Things' and 'Big Data', holds promises for more efficient management of energy, water and transport services in large cities - all leading to a rejuvenation of cities with sustainable economic development and a better quality of life [3]. This enforces us to do real-time analytics to take real-time decisions. Hadoop ecosystem tools have made it feasible to collect, store and process such data and harness it for real-time analytics. As a result, more and more companies are looking to include non-traditional yet potentially very valuable data with their traditional enterprise data in their business intelligence analysis.

Even though there are umpteen number of definitions for big data available, the definition of big data in our terminology is **"It is the data that is too large, complex, and dynamic in a way that it is impractical for any conventional hardware and/or software tools and systems to manage and process in *timely manner and scalable fashion"***.

A. 5 V's of Big Data

To understand the characteristics of big data, it is often described using five 'V's [4]: Volume, Velocity, Variety, Veracity and Value.

Fig. 1 5 V's of Big Data.

Volume: It refers to the vast amounts of data generated in the form of emails, twitter messages, photos, video clips, sensor data etc. In no time, we may be talking about Zettabytes or Brontobytes. On Facebook alone, people send 10 billion messages per day, average daily likes 4.5 billion times and upload 350 million new pictures each and every day [5]. Akamai analyses 75 million events a day to target online ads; Walmart handles 1 million customer transactions every single hour. Smart meters generate similar data volumes, compounding the problem. In fact, if we take all the data generated in the world upto 2008, the same amount of data may be generated every minute in the near future!

Variety: It refers to the different types of data we are generating every day. In the past, we were mostly focusing on the structured data that neatly fits into tables of relational databases. Now, in fact, 80% of the world's data is unstructured (social media updates, e-mail messages, logs, videos, photos, audio files etc.), and therefore can't be processed easily using traditional database technologies.

Velocity: It refers to the speed at which new data is generated and the data has be processed, and analyzed. Every minute of every day, we upload 100 hours of video on Youtube, send over 200 million emails and send 300,000 tweets. Even at 140 characters per tweet, the high velocity of Twitter data ensures large volumes (over 8 TB per day). Also, consider the speed at which credit card transactions are checked for fraudulent activities; the milliseconds it takes trading systems to analyze and pick up signals that trigger decisions to buy or sell shares. Since we can't store this humongous data into databases, we need to analyze the data while it is being generated, without even putting it into databases.

Veracity: It refers to the uncertainty or trustworthiness of the data. With many forms of big data, quality and accuracy are less controllable. For example, consider Twitter posts with hash tags, abbreviations, typos and colloquial speech as well as the reliability and accuracy of content. The data is worthless if it's not accurate which is particularly true in programmes that involve automated decision-making.

Value: It is all well and good having access to big data but unless we can use it to take analytical decisions to improve the business strategies, it is useless. Typically,

there may be good amount of information hidden in the data; the challenge is to identify what is valuable and then transforming and extracting that data for analysis. When big data is distilled and analyzed, enterprises can develop a more thorough and insightful understanding of their business, which can lead to enhanced productivity, a stronger competitive position and greater innovation. Hence, by all means, 'value' is the most important **'V'** of big data.

B. Challenges with Big Data:

The growth of data in different dimensions brings out lots of challenges as well as opportunities for organizations. With growing data volume and velocity, it is essential that real-time information useful for business decisions can be extracted, otherwise the business risks being swamped by a data deluge in today's hyper-competitive business environment. But the challenge is going through the sheer volumes of data and accessing the level of detail needed, all at a high speed. The challenge only grows as the degree of granularity increases. Hence, it is ambitious to be able to use real-time data for real-time decision-making to become a real-time business.

In addition to the increasing velocity and varieties of data, data flows can be highly inconsistent with periodic peaks. Daily, seasonal and event-triggered peak data loads such as social media data can be challenging to manage. It becomes even more complicated when unstructured data is involved because now a days data comes in variety of form of emails, photos, videos, monitoring devices, PDFs, audio etc. These varieties of unstructured data create problems for storage, mining and analysis. Also, in the domains such as internet, finance and smart grids, the number of potential training examples is extremely large, making single-machine processing in-feasible. This enforces us to look for the database management systems that can deal with non-relational, schema-less, distributed databases.

Data visualization can communicate trends and outliers much faster than tables containing numbers and text. But, it becomes difficult task when dealing with extremely large amounts of information or a large variety of categories of information. Also, outliers typically represent about 1 to 5 percent of data, but when you are working with massive amounts of data, viewing 1 to 5 percent of the data is rather difficult. Hence, it is non-trivial task to represent those points without getting into visualization issues. On the other hand, when you are looking for events of a certain type within large amount of data, you can expect events of this type to occur, even if the data is completely random. Also, the number of occurrences of these events will grow as the size of the data grows. These occurrences are 'spurious' in the sense that they have no cause. Other than that random data will always have some number of unusual features that look significant but aren't.

When we are dealing with streaming data, much of this data is of no interest, and it can be filtered and compressed by orders of magnitude [6]. One challenge is to define these filters in such a way that they do not discard useful information. For example, suppose that one sensor reading differs substantially from the rest, it may be due to the sensor being faulty or it may be the reading that needs attention. But how can we be sure that it is not an artifact that deserves attention? In addition, the data collected by these sensors most often are spatially and temporally correlated (e.g.,

traffic sensors on the same road segment). We need techniques for data reduction that can intelligently process this raw data to a size that its users can handle without missing the needle in the haystack. Furthermore, we require 'on-line' analysis techniques that can process such streaming data on the fly, since we cannot afford to store first and reduce afterward.

Even if you can find and analyze data quickly and put it in the proper context, the value of data for decision-making purposes will be jeopardized if the data is not accurate or timely. This is a challenge even with small data sets. But, when we consider humongous data, it becomes even more challenging. Big data is generally characterized by high dimensionality and large sample size. These two features may raise challenges [7] as: (i) noise accumulation, spurious correlations, and incidental homogeneity; (ii) issues such as heavy computational cost and algorithmic instability; (iii) typically aggregated from multiple sources at different time points using different technologies. This creates issues of heterogeneity, experimental variations, and statistical biases, and requires us to develop more adaptive and robust procedures.

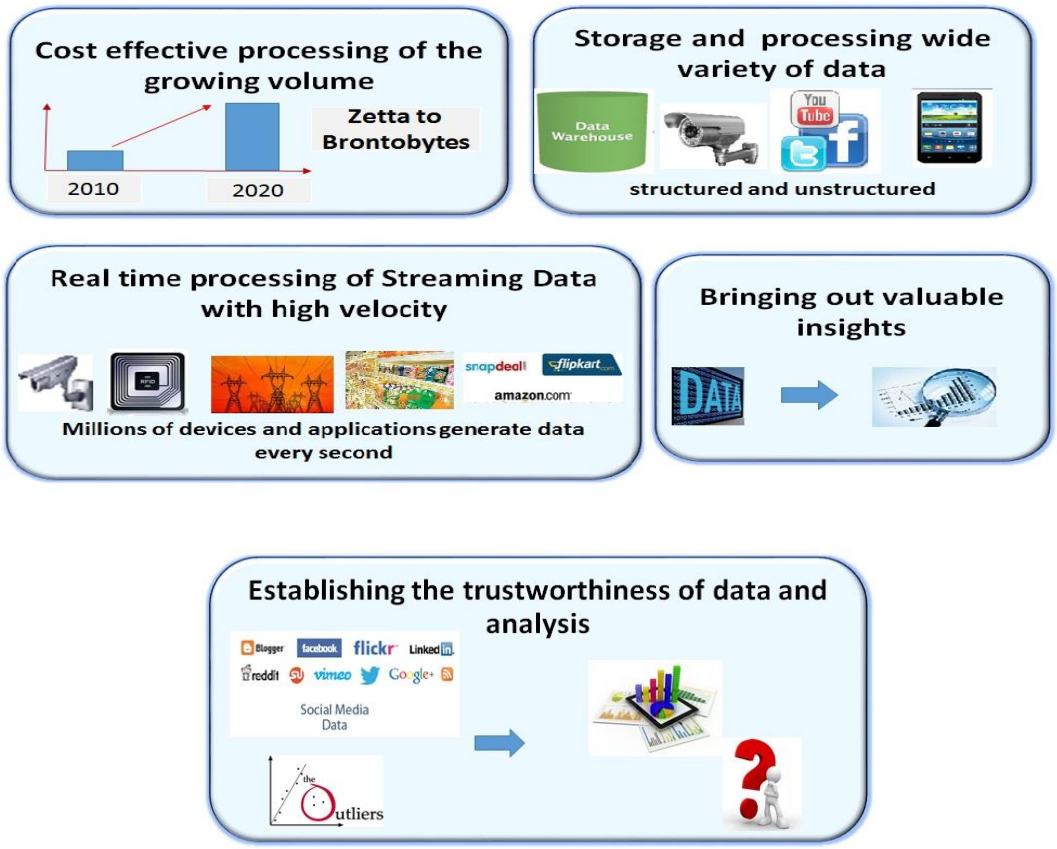

Fig. 2 Challenges with Big Data.

In the next three sections, we will discuss how to overcome the challenges that arise due to Volume, Variety and Velocity. In the section 2, we discuss about how Hadoop can solve the problem of processing humongous data. In the section 3, we

discuss about how NoSQL databases can be used to solve the problems arise due to variety. In the section 4, we discuss about the ways to handle the problems arise due to velocity. In the section 5, we briefly discuss about the analytics of big data. In the section 6, we discuss several use cases of big data.

2. Basic modules of Hadoop

Hadoop is a high-throughput system which can crunch a huge volume of data using a distributed parallel processing paradigm called MapReduce. *Hadoop ecosystem* is the term used for a family of related projects that fall under the umbrella of infrastructure for distributed computing and large-scale data processing. The open source Hadoop framework and ecosystem technologies are critical components in a growing number of big data initiatives for both storing and processing data at dramatically lower costs.

A. Basic Components of Hadoop

There are two basic components at the core of Apache Hadoop-1: the Hadoop Distributed File System (HDFS), and the MapReduce parallel processing framework.

1. *HDFS:* HDFS is the storage component of Hadoop. It's a distributed file-system which is modeled after the Google File System (GFS) paper. HDFS is a fault tolerant and self-healing distributed file system designed to turn a cluster of industry standard servers into a massively scalable pool of storage. It is developed specifically for large-scale data processing workloads where scalability, flexibility and throughput are critical.

HDFS is optimized for high throughput and works best when reading and writing large files (gigabytes and larger). To support this throughput HDFS leverages unusually large (for a file-system) block sizes and data locality optimization to reduce network input/output (I/O). Scalability and availability are also key traits of HDFS, achieved in part due to data replication and fault tolerance. HDFS replicates files for a configured number of times, is tolerant of both software and hardware failure, and automatically re-replicates data blocks on nodes that have failed.

2. *MapReduce:* MapReduce is a massively scalable, parallel processing framework that works in tandem with HDFS. With MapReduce and Hadoop, computations are executed at the location of the data, rather than moving data to the computing location i.e. data storage and computation coexist on the same physical nodes in the cluster [8]. MapReduce processes exceedingly large amounts of data without being affected by traditional bottlenecks like network bandwidth by taking advantage of this data proximity. MapReduce is primarily designed for *batch processing* over large datasets.

The MapReduce model simplifies parallel processing by abstracting away the complexities involved in working with distributed systems, such as computational parallelization, work distribution, and dealing with unreliable hardware and software. With this abstraction, MapReduce allows the programmer to focus on addressing bus*iness needs, rather than getting tangled up in distributed system complications.*

MapReduce decomposes work submitted by a client into small parallelized map and reduce tasks. The map and reduce constructs used in MapReduce are borrowed

from those found in the Lisp functional programming language, and use a shared-nothing model to remove any parallel execution interdependencies that could add unwanted synchronization points or state sharing. Each task has key-value pairs as input and output, the types of which may be chosen by the programmer.

B. Hadoop Ecosystem

Here, we describe a few important Hadoop ecosystem tools and their functionalities. But, this is not an exhaustive list of Hadoop ecosystem tools. An example of Hadoop ecosystem is shown in figure **3**. This information is largely taken from websites provided by Apache software foundation.

- **Apache Pig** is a high-level procedural language for querying large semi-structured data sets using Hadoop and the Map/Reduce Platform. Pig simplifies the use of Hadoop by allowing SQL-like queries to run on distributed dataset.
- **Apache Hive** is a data warehousing infrastructure based on the Hadoop. It provides a simple query language called HiveQL, which is based on SQL. Hive has three main functions: data summarization, query and analysis. Hive automatically translates SQL-like queries into Map/Reduce jobs that run Hadoop cluster.
- **Apache Sqoop** is a tool designed for transferring bulk data between Apache Hadoop and structured datastores such as relational databases or data warehouses.
- **Apache Flume** is a distributed system to reliably collect, aggregate and move large amounts of log data from many different sources to a centralized data store.
- **Apache HBase** is a column-oriented, non-relational, distributed database management system that runs on top of HDFS. HBase, which is modeled after Google's BigTable, can handle massive data tables containing billions of rows and millions of columns.
- **Apache ZooKeeper** provides operational services for a Hadoop cluster. ZooKeeper provides a distributed configuration service, a synchronization service and a naming registry for distributed systems. Distributed applications use Zookeeper to store and mediate updates to important configuration information.
- **Apache Drill** is a low latency SQL query engine that supports data-intensive distributed applications for interactive analysis of large-scale datasets.
- **Apache Storm** is a distributed realtime computation system to reliably process unbounded streams of data, doing for realtime processing what Hadoop do for batch processing.
- **Apache Mahout** is a library of scalable machine-learning algorithms, implemented on top of Apache Hadoop using the MapReduce paradigm.
- **Apache Oozie** is a server based Workflow Engine specialized in running workflow jobs with actions that run Hadoop Map/Reduce, Pig jobs and other. Oozie workflow is a collection of actions arranged in DAG and enable us to process large amount of data quickly without running out of memory.

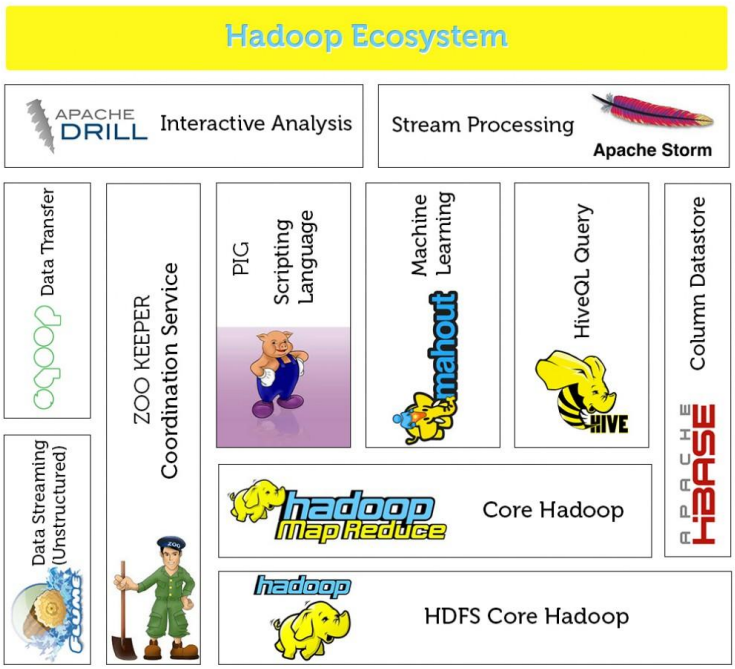

Fig. 3 Hadoop Ecosystem.

C. Strengths of Hadoop ecosystem

The earlier big data solutions include data sharding i.e. breaking the database down into smaller chunks called 'shards' and spreading them across a number of distributed servers. Data sharding has many shortcomings: a) Reliability; b) Writing sharding Code is difficult; c) No automated way to perform load balancing; d) Shards are not synchronously replicated. On the other hand, data for distributed systems is stored on a SAN (storage-area network) and it is copied to the computing nodes at the time of computing. But, SANs are very expensive. Also, Programming for traditional distributed systems is very complex and it has the following shortcomings: a) Data exchange requires synchronization; b) Bandwidth is limited; c) Temporal dependencies are complicated; d) It is difficult to deal with partial failures of the system.

We can overcome all the shortcomings discussed above using Hadoop ecosystem. The strengths of Hadoop ecosystem are given below:

- **Scalability:** New nodes can be added as needed.
- **Cost effectiveness:** It brings massively parallel computing to commodity servers that results in sizable decrease in the cost per terabyte of storage, which in turn makes it affordable to model all the data.
- **Flexibility:** It can absorb any type of data whether it is structured or unstructured from variety of sources and enable deeper analyses than any system can provide.
- **Fault tolerance:** When a node is lost, it redirects work to another location of the data and continues processing without missing a beat.

- **Reduction in network communications:** Computations are executed at the location of the data, rather than moving data to the computing location; data storage and computation coexist on the same physical nodes in the cluster.
- **Abstraction:** Hide system-level details from the application developer.

D. Shortcomings of Hadoop-1

Apache Hadoop-1 had the following issues:

1. It has only one NameNode, which manage the whole cluster. This node is a single point of failure.
2. The MapReduce paradigm can be applied to only a limited type of tasks. There are no other models (other than MapReduce) of data processing.
3. Resources of a cluster are not utilized in the most effective way.

E. Improvements in Hadoop-2

There are two major improvements in Hadoop-2 to overcome the shortcomings of Hadoop-1

1. HDFS Federation: multiple, redundant namenodes acting together
2. YARN (MRv2): Here, the fundamental idea of is to split up the two major functionalities of the JobTracker, resource management and job scheduling/monitoring, into separate daemons. The idea is to have a global ResourceManager (RM) and per-application ApplicationMaster (AM). YARN (Yet Another Resource Negotiator) is a resource-management platform responsible for managing compute resources in clusters and using them for scheduling of users' applications. It enables user to interact with all the data in multiple ways simultaneously, making Hadoop a true multi-use data platform.

By separating the data processing engine of Hadoop (MapReduce) from the resource management, Hadoop-2 enables many different processing engines can operate simultaneously across a Hadoop cluster, on the same data, at the same time. Hadoop-2 architecture is shown in figure **4**.

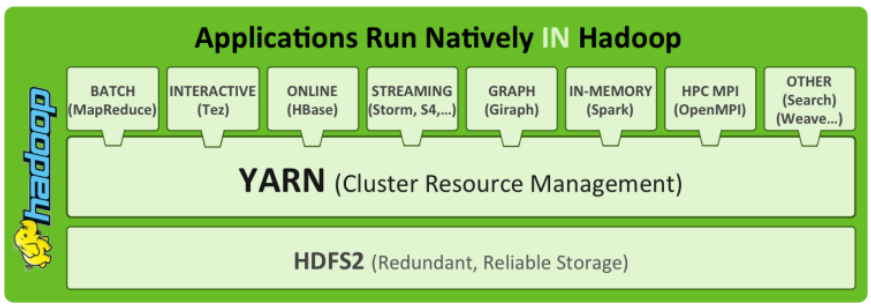

Fig. 4 Hadoop-2 Architecture (Source: Hortonworks).

The figure **5** compares Hadoop-1 and Hadoop-2 architectures.

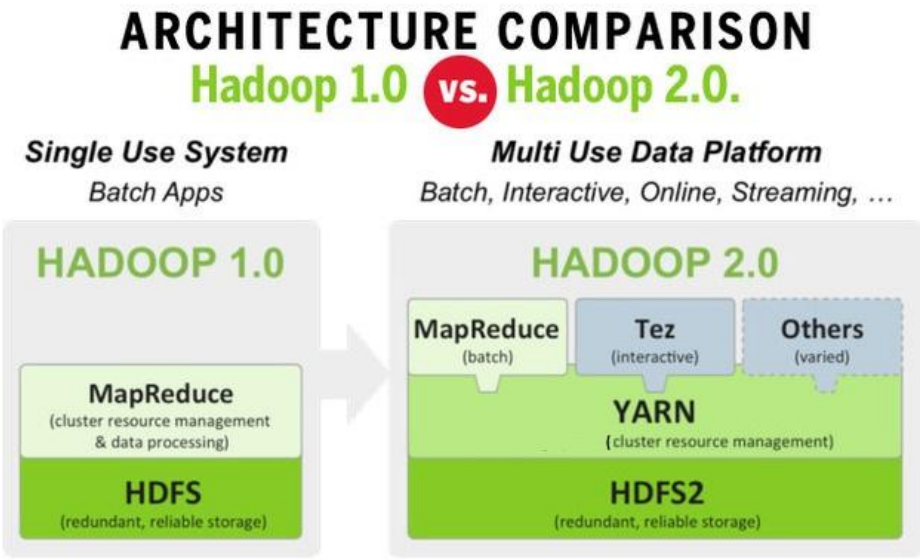

Fig. 5 Hadoop Architecture Comparison (Source: Hortonworks)

Hadoop provides three installation choices:

a. Local mode: This is an unzip and run mode to get you started right away where all components of Hadoop run within the same JVM
b. Pseudo distributed mode: This mode will run different components of Hadoop as different Java processors, but within a single machine
c. Fully Distributed mode: This mode is obviously the only one that can scale Hadoop across a cluster of machines

Many people generally prefer the pseudo-distributed mode even when using examples on a single host, because everything done in the pseudo-distributed mode is almost identical to how it works on a much larger cluster.

F. Word Count Example

Applications typically implement the Mapper and Reducer interfaces to provide the map and reduce methods. Mapper maps input key/value pairs to a set of intermediate key/value pairs that need not to be of the same type as the input. A given input pair may map to zero or many output pairs. The number of maps is usually driven by the total size of the inputs, that is, the total number of blocks of the input files. The right level of parallelism for maps seems to be around 10-100 maps per-node, although it may be upto 300 maps for very CPU-light map tasks [9].

Reducer reduces a set of intermediate values which share a key to a smaller set of values. Reducer has 3 primary phases: shuffle, sort and reduce. In the shuffle phase, the framework fetches the relevant partition of the output of all the mappers, via HTTP. In the sorting phase, framework groups reducer inputs by keys (since different mappers may have output the same key). The shuffle and sort phases occur simultaneously; while map-outputs are being fetched they are merged. MapReduce

makes the guarantee that the input to every reducer is sorted by key. During the reduce phase, the reduce function is invoked for each key in the sorted output. The output of this phase is written directly to the output file-system, typically HDFS. The right number of reduces seems to be 0.95 or 1.75 multiplied by (<no. of nodes> * mapred.tasktracker.reduce.tasks.maximum)[9]. A MapReduce process for simple word count is shown in figure **6**.

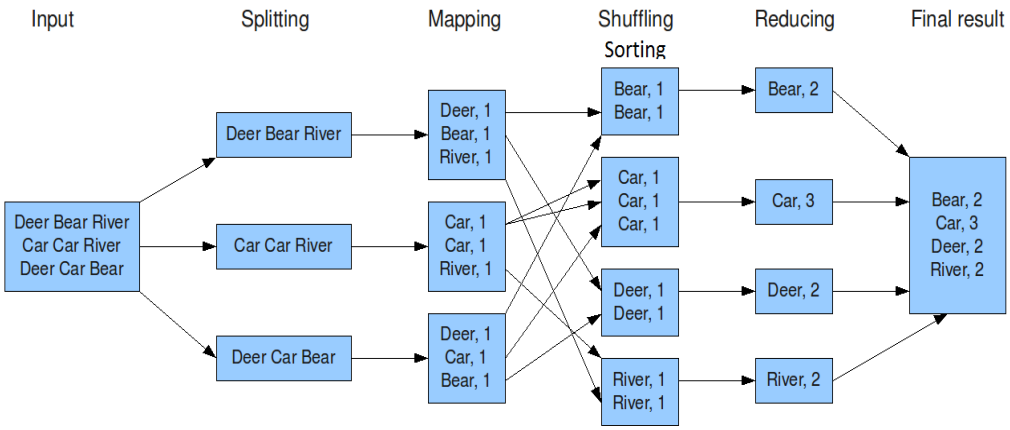

Fig. 6 Simple Word Count Process in MapReduce

3. NoSQL

NoSQL (Not only SQL) encompasses a wide variety of different database technologies that are non-relational, schema-less, distributed, open-source and horizontally scalable. They are designed to cope up with big data and real-time web applications that require analysis of extremely high-volume, disparate data types. Many of NoSQL systems do not attempt to provide atomicity, consistency, isolation and durability guarantees, contrary to the prevailing practice among relational database systems. Relational databases, on the other hand, neither designed to cope with the scale and agility challenges from modern applications, nor to take advantage of the cheap storage and distributed processing power available today.

There are around 150 different NoSQL databases[10]. There have been various approaches to classify NoSQL databases, each with different categories and subcategories. Nevertheless, the most basic classification that most would agree is the one that is based on the data model. A few of them and their prototypes are[11]:

1. Column : Hbase, Cassandra, Accumulo, Hypertable
2. Document : MongoDB, CouchDB, RavenDB, RethinkDB, Terrastore
3. Key-value : Dynamo, Riak, Redis, MemcacheDB, Voldemort, Scalaris, BerkeleyDB, SimpleDB
4. Graph : Neo4J, AllegroGraph, Infinite Graph, HyperGraphDB
5. Multimodel: CortexDB, AlchemyDB

A. Brief Description of Various NoSQL Databases

Graph: It is designed for data whose relations are well represented as a graph i.e. elements interconnected with an undetermined number of relations between them. A graph database is essentially a collection of nodes and edges. Each node represents an entity and each edge represents a connection or relationship between two nodes. Every node in a graph database is defined by a unique identifier, a set of outgoing edges and/or incoming edges and a set of properties expressed as key/value pairs. Each edge is defined by a unique identifier, a starting-place and/or ending-place node and a set of properties. This kind of data could be social relations, public transport links, road maps or network topologies, for example. Graph databases are well-suited for mining the data from social media. Graph databases are also useful for working with data in disciplines which involve complex relationships and dynamic schema, such as supply chain management, biology and recommender systems [12].

Column: In this, data is stored in cells grouped in columns of data rather than as rows of data. Columns are logically grouped into column families. Column families can contain a virtually unlimited number of columns that can be created at run-time or at the definition of the schema. Read and write is done using columns rather than rows. In comparison, most relational DBMS store data in rows, the benefit of storing data in columns, is fast search/ access and data aggregation. Relational databases store a single row as a continuous disk entry. Different rows are stored in different places on disk while Columnar databases store all the cells corresponding to a column as a continuous disk entry thus makes the search/access faster. For example, Cassandra is capable of handling business applications that require massive scalability, continuous availability, high performance, strong security, and operational simplicity. Companies such as Netflix, Sky, SoundCloud, Healthx, GoDaddy, eBay etc. are using Cassandra [13].

Document: The central concept of a document store is the notion of a "document" and the data in the document is a collection of key value pairs. XML, YAML, and JSON as well as binary forms like BSON are some common standard encodings. One key difference between a key-value store and a document store is that the latter embeds attribute meta-data associated with stored content, which essentially provides a way to query the data based on the contents. For example, MongoDB used for operational intelligence and real time analytics. Bosch has built its Internet of Things suite on MongoDB, bringing the power of big data to a new range of Industrial Internet applications including manufacturing, automotive, retail, energy and many others [14].

Key Value Store: The key-value model is one of the simplest non-trivial data models in schema-less format. The key can be synthetic or auto-generated while the value can be String, JSON, BLOB (basic large object) etc. Basically, it uses a hash table in which there exists a unique key and a pointer to a particular item of data. For example, Riak can be used in applications which needs session stores in e-Commerce sites. Companies such as Best Buy, Copious, Ideel, Shopzilla uses Riak [15].

Figure **7** gives the comparison of NoSQL databases w.r.t complexity v.s. scalability and figure **8** gives the comparison of NoSQL databases by Ben Scofield

[16].

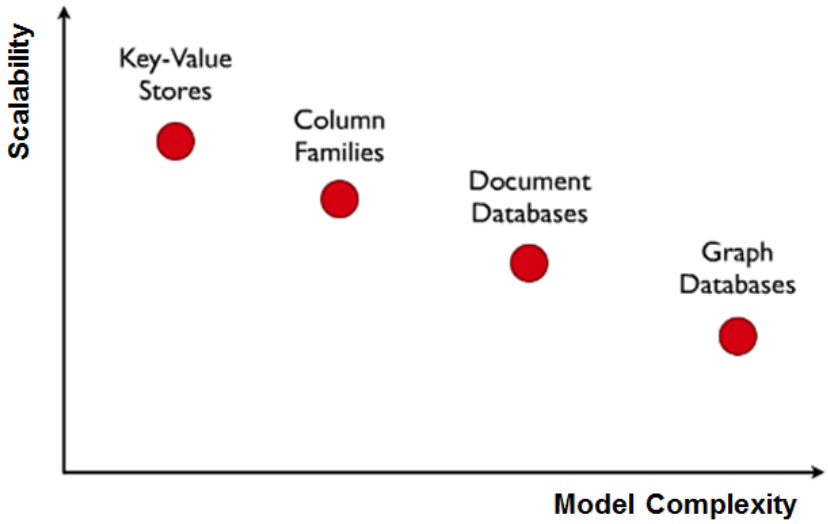

Fig. 7 Comparison of NoSQL databases Complexity Vs. Scalability

	Performance	Scalability	Flexibility	Complexity	Functionality
Key-Value Stores	high	high	high	none	variable (none)
Column stores	high	high	moderate	low	minimal
Document stores	high	variable (high)	high	low	variable (low)
Graph databases	variable	variable	high	high	graph theory
Relational databases	variable	variable	low	moderate	relational algebra

Fig. 8 Comparison NoSQL databases by Ben Scofield.

B. Brewer's CAP Theorem

As the size of data grew immensely, it made indispensable to find more scalable solutions than the so far existing relational (ACID) databases. As a result new principles were developed, summed up under the BASE paradigm (basically available, soft state, eventual consistency).

In 2000, Eric Brewer conjectured that a distributed system cannot simultaneously provide all three of the following desirable properties:

a) **Consistency:** A read sees all previously completed writes.
b) **Availability:** A guarantee that every request receives a response about whether it succeeded or failed.
c) **Partition tolerance:** Guaranteed properties are maintained even when network failures prevent some machines from communicating with others.

The CAP theorem states that "One can have at most two of these properties for any shared-data system". But, the trade-off between Consistency and Availability only has to be considered when the network is partitioned. One can interpret this fact in such a way that the system should forfeit Partition Tolerance as long as there is no

partition, and as soon as a network partition occurs it needs to switch its strategy and choose a trade-off between Consistency and Availability.

Figure **9** shows the classification of NoSQL databases according to the properties that they satisfy in CAP theorem.

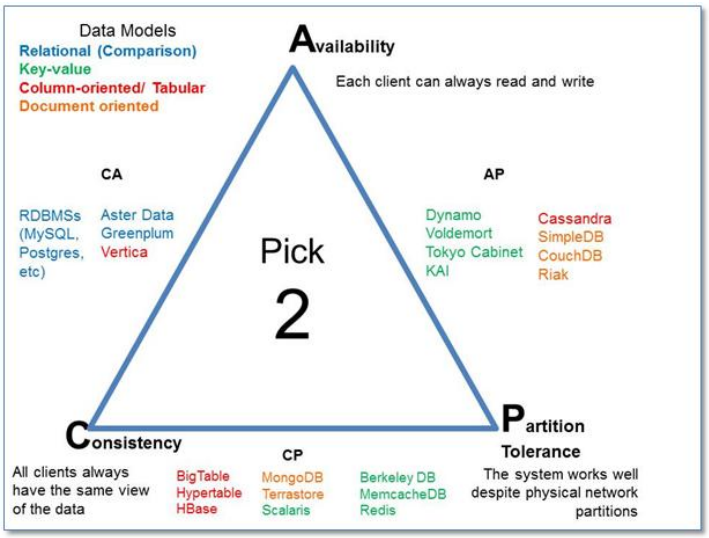

Fig. 9 Classification of NoSQL databases based on CAP Theorem [17].

4. Streaming Data

Due to electronics/digital revolution, many applications are producing large volumes of data continuously with very high velocity in a myriad of formats. Figure **10** gives an example for streaming data. Such kind of data is called streaming data. Examples of data streams include computer network traffic, phone conversations, ATM transactions, web searches, and sensor data etc. The data are temporally ordered, fast changing, massive, and potentially infinite. With the advent of *smart cities* concept and *IOT* (*Internet of Things*), many applications require continuous analysis of data as it is captured, in real-time to take immediate actions. To discover knowledge from such data, it is necessary to develop single-scan, on-line, multilevel, multidimensional stream processing and analysis methods.

There are a number of aspects of streaming data that set it apart from other kinds of data. All of the components of real-time systems share three key features that allow them to operate in a real-time streaming environment: high availability, low latency, and horizontal scalability [18]. The data is always being generated. This has a few effects on the design of any collection and analysis system. Firstly, the collection itself needs to be very robust. Downtime for the primary collection system means that data is permanently lost. Secondly, the fact that the data is always coming in means that the system needs to be able to keep up with the data. If 2 minutes are required to process 1 minute of data, the system will not be real time for very long. Eventually, the problem will be so bad that some data will have to be dropped to allow the system

to catch up. In practice it is not enough to have a system that can merely "keep up" with data in real time. It needs to be able to process data far more quickly than real time.

Thirdly, in the streaming setting, the data can usually be seen a single time. More commonly, there is no previous data to perform an analysis upon. In this case, the streaming system must attempt to deal with the data at its natural cardinality. This is very challenging task both in terms of processing and in terms of storage. Performing analysis on a large data set necessarily takes time to process anything that involves a large number of different states. It also requires a linear amount of space to store information about each different state. Unlike batch processing, storage space is much more restricted because it must generally use very fast main memory storage instead of the much slower tertiary storage of hard drives. Even with the availability of high performance Solid State Drives (SSDs), they are still orders of magnitude slower than memory access.

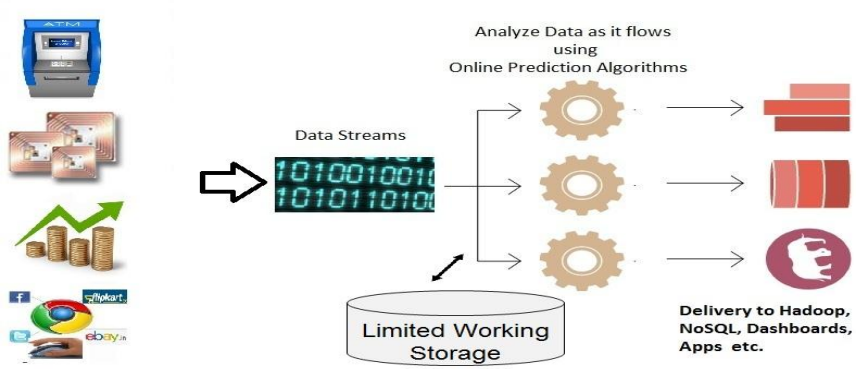

Fig. 10 Streaming Data Analytics.

A. In-Memory Data Processing

Retrieving data from disk storage is the slowest part of data processing. And the more data you need to work with, the more the retrieval step slows down the analytics process. In-memory technologies have become increasingly important as companies see the need to manage and analyze growing volumes of potentially disparate and real-time data. In-memory processing overcomes this hurdle completely, because for analytical purposes all the relevant data is loaded into RAM and therefore does not have to be accessed from disk storage. For example, data streams from the Internet of things can be captured and processed in-memory to predict future events or recognize current events that require action. For real-time streaming data, in-memory processing can enable mathematical computations to be performed in RAM rather than the disk, enabling processing thousands of times faster than data access from disk.

B. Apache Storm

Apache Storm is a distributed real-time computation system to reliably process unbounded streams of data, doing for real-time processing what Hadoop do for batch processing. Storm has many use cases: real-time analytics, online machine learning,

continuous computation, distributed RPC, ETL, and more [19].

C. Apache Spark

Apache Spark is a data analytics cluster computing framework that provides primitives for in-memory cluster computing and allows user programs to load data into a cluster's memory and query it repeatedly, making it well suited to machine learning algorithms [20]. Apache Spark runs on Hadoop, Mesos, standalone, or in the cloud. It can access diverse data sources including HDFS, Cassandra, HBase, S3. Also, Spark powers a stack of high-level tools including Spark SQL, MLlib for machine learning, GraphX, and Spark Streaming. One can combine these frameworks seamlessly in the same application.

Both Apache Storm and Apache Spark can be crucial solutions that can surpass the challenges bring forth by streaming data.

5. Analytics

In the article "The Age of Big Data" by Steve Lohr [21] which appeared in New York times on February 11, 2012, Steve Lohr referred Brynjolfsson et al. research work in which they studied 179 large companies and found that those adopting "data-driven decision making" achieved productivity gains that were 5% to 6% higher.

Arguably, the Big Data problem has existed for a very long time. The difference now is that we have finally reached the point where we can do something with the data - such as asking interesting questions of it, making more informed and faster business decisions, and providing services that allow consumers and businesses to leverage what is happening around them right now. Also, it is becoming increasingly necessary for the companies to take decisions based on data analytics as Peter Sondergaard of the Gartner Group quotes "Information is the oil of the 21st century, and analytics is the combustion engine".

A. Types of Analytics

From a taxonomical view, there are three main categories under analytics: descriptive, predictive and prescriptive. Figure **11** illustrates this simple taxonomy of analytics [22].

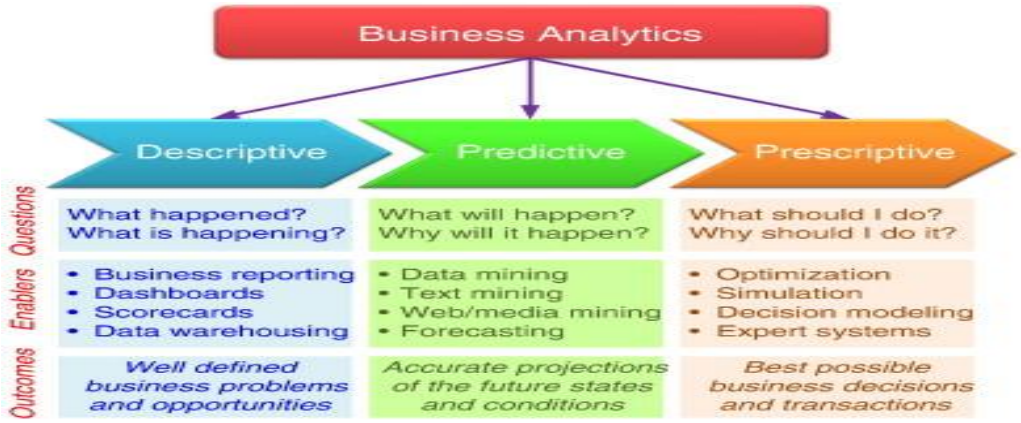

Fig. 11 A simple taxonomy of business analytics [22].

a. ***Descriptive analytics:*** Descriptive analytics looks at historical data to find the reasons behind past success or failure. Business intelligence is the world of descriptive analytics i.e. retrospective analysis that provides a rear-view mirror view of the business-reporting on what happened and what is currently happening. Common examples of descriptive analytics are management reports providing information regarding sales, customers, operations, finance and to find correlations between the various variables.

b. ***Predictive analytics:*** Predictive analytics turns data into valuable, actionable information. Predictive analytics uses data to determine the probable future outcome of an event or a likelihood of a situation occurring. Predictive analytics encompasses a variety of statistical techniques, modeling, machine learning and data mining that analyze current and historical facts to make predictions about future events. Predictive analytics models capture relationships among many factors to asses risks or opportunities for future with a particular set of conditions. By successfully applying predictive analytics, the businesses can effectively make use of big data for their benefit.

c. ***Prescriptive analytics:*** Prescriptive analytics goes beyond predicting future outcomes by suggesting actions to benefit from the predictions and showing the decision maker the implications of each decision option. Prescriptive analytics not only foresees what will happen and when it will happen, but also why it will happen and provides recommendations how to act upon it in order to take advantage of the predictions.

Figure **12** shows few differences between these three taxonomies of analytics by taking retail analytics example.

Descriptive Analytics	Predictive Analytics	Prescriptive Analytics
Can Answer questions like: What is the attrition rate in last six months? Which customers I have lost?	Can Answer questions like: Why the attrition rate increased in last six months? Which customers are most likely to attrite?	Can Answer questions like: Which customers should I target to retain? What can I offer before the customers even realizes the need?
Hindsight	Insight	Foresight
Simple Statistics, BI Tools, Dash Boards etc.	Data Mining, Machine Learning	Modeling and game theory, Simulation, Optimization

Fig. 12 Comparison of three different taxonomies of analytics.

B. Predictive Analytics

Predictive analytics is the practice of extracting information from existing data sets in order to determine patterns and predict future outcomes and trends. Predictive

analytics enable organizations to use big data (both batch and real-time) for taking insightful decisions. It uses many machine learning and data mining algorithms.

Machine learning deals with the construction and study of algorithms that can learn from data and improves its performance at some task through experience. Machine Learning algorithms are the core algorithms in Data Mining.

Data mining is the core stage of the knowledge discovery process that is aimed at the extraction of interesting, nontrivial, implicit, previously unknown and potentially useful information from data in large data sets. Machine learning algorithms have many potential applications such as online recommendations, ad targeting, virtual assistants, demand forecasting, fraud detection, spam filters, speech recognition, bio-informatics, machine translation, bio-metrics, medical diagnosis etc.

Even though there are several Machine learning algorithms available, it is extremely non-trivial task to choose and use these algorithms for any given task. Figure **13** shows steps involved in applying machine learning or data mining algorithms for knowledge discovery task.

a. **Understanding:** The first step is understanding the requirements. One need to have a clear understanding about the application domain, relevant prior knowledge and objectives. Based the objective, one has to formulate the problem as Machine Learning or Data mining task. Some common data mining tasks are classification, clustering, regression, association rule discovery, sequential pattern discovery and deviation detection.

b. **Selection of data set:** One should select a target data set or subset of data on which one needs to perform data analysis to extract useful knowledge. It is important when creating this target data that the data analyst understand the domain, the end user's needs, and what the data mining task might be.

c. **Data cleaning:** In general, data mining algorithms assume that data is available, relevant, adequate, and clean. This step includes handling of noise and irrelevant data in the large data set. This is a very important pre-processing step because outcome would be dependent on the quality of selected data. As part of this step de-duplication, handling of missing values, removal of unnecessary data fields, standardization and so on will be take care.

d. **Data transformation:** With the help of dimensionality reduction or transformation methods, the number of effective variables is reduced and only useful features are selected to depict data more efficiently based on the goal of the task.

e. **Selection & Application of data mining algorithm:** Appropriate method(s) is(are) to be selected for looking for patterns from the data. One need to decide the model and parameters that might be appropriate for the method. Then one need to apply the method on the selected data to extract patterns from the data.

f. **Pattern evaluation:** One need to evaluate and interpret the mined patterns and relationships. If the patterns evaluated are not useful, then the whole process can be iteratively applied from any of the previous steps.

g. **Consolidation:** The knowledge discovered is consolidated and represented to the user in a simple and easy to understand format. Mostly, visualization techniques are being used to make users understand and interpret information.

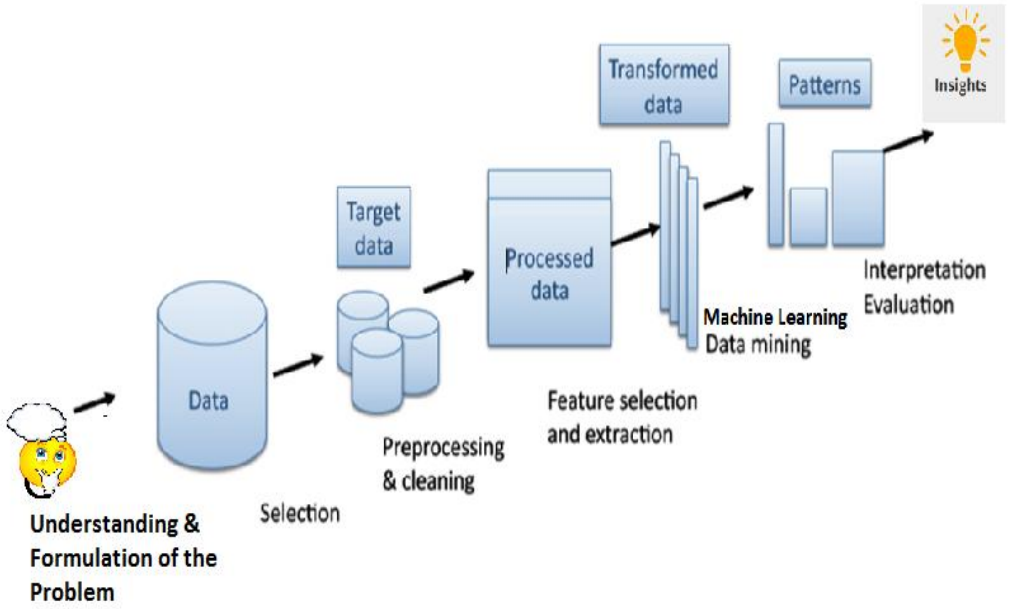

Fig. 13 Knowledge Discovery Process.

Here, we describe the different tasks that data mining or Machine Learning algorithms can perform and the algorithms used to accomplish that task.

a. **Classification:** Predicting the instance class using the model learned from pre-labeled (classified) instances. Some of the algorithms used: Decision Trees, SVM, Multilayer Perceptron, RBF networks, Naive Bayes etc.

b. **Clustering:** Finding 'natural' groups of instances given unlabeled data in such a way that the objects are 'similar' within the group and 'dissimilar' across the groups. Some of the algorithms used: K-means, K-medoid, DBSCAN, SOM, GMM, Fuzzy C-means, Latent Dirichlet Allocation etc.

c. **Regression:** Estimating the relationships among variables for prediction. Some of the algorithms used: Linear, Multiple linear, Logistic, Poisson etc.

d. **Association Rule Discovery:** Mining any rule of the form $X \rightarrow Y$, where X and Y are sets of data items. Some of the algorithms used: Apriori, FP-tree, Rapid Association Rule Mining etc.

e. **Sequential Pattern Mining:** Finding all frequent sequences. Some of the algorithms used: Generalized Sequential Patterns (GSP), Sequential PAttern Discovery using Equivalence classes (SPADE).

f. **Temporal Pattern Recognition:** Segmenting sequences that recurs frequently in the whole temporal sequence. Some of the algorithms used: Dynamic Bayesian networks, Hidden Markov Models etc.

g. **Structured Prediction:** Predicting the output labels jointly by considering the constraints and exploiting the correlations in the output spaces like sequences or trees. Some of the algorithms used: Conditional Random Fields, Structured SVMs, Markov logic networks etc.

h. **Dimensionality Reduction:** Choosing a subset of all the features (Feature

selection) or creating new features by combining existing ones. Some of the algorithms used: Principal Components Analysis (PCA), linear discriminant analysis (LDA), Locally Linear Embedding, Laplacian Eigenmaps, Latent Semantic Indexing etc.

C. Challenges for Machine Learning on Big Data

Here, we describe few challenges and the ways to overcome those challenges while applying machine learning algorithms on big data for analytics.

a) Large number of data instances: In many domains, the number of potential training examples is extremely large, making single-machine processing in-feasible.

i) One way to effectively process such datasets is to use MapReduce framework or DryadLINQ. They use distributed processing on clusters with large number of machines to accomplish the task. But, only a few machine learning algorithms can be put into MapReduce framework. Shih-Ying Chen et. al. used MapReduce framework for finding association rules from large data sets [23]. In some of the applications like recommender systems, where problems involve matrices with millions of rows and columns, and billions of instances, distributed algorithms for matrix factorization can help in achieving reasonable performance [24][25].

ii) When the data volume is too large to be handled by a single classifier, an ensemble or meta-ensemble methods (such as Random Forests, Stacking) can let each classifier process a partition of the data and then combine their results [26].

iii) In case of supervised learning algorithms, they require plenty of labeled data. But, creating labeled data is a arduous task. Hence, we require semi-supervised learning algorithms which aim to scale up to large datasets to really achieve the goals. Instances of such approach are graph-based approaches(Markov random walks on graphs), change of representation based approaches and margin based approaches (TSVM) [27].

b) Streaming data processing: In general machine learning algorithms assume that the whole data is available for learning the optimal parameters i.e. for generalization. But, in case of streaming data this assumption is not true. A large number of techniques have been proposed to address the research issues of analyzing rapidly arrived data streams in real time which can be classified into four different categories [28][29].

1. Two-phase techniques: CluStream, HPStream
2. Hoeffding bound-based techniques: Very Fast K-Means (VFKM), Very Fast Decision Trees (VFDT)
3. Symbolic approximation-based techniques: Symbolic Aggregate approXimation (SAX)
4. Granularity-based techniques: one-look clustering algorithm (LWC), Light Weight K-nearest neighbor Classification (LWclass)

c) High Input Dimensionality: Machine learning and data mining tasks involving natural language, images, or video can easily have input dimensionality of 10^6 or higher. Feature Subset Ensemble Approaches are particularly useful for high-dimensional datasets because increased classification accuracy can be achieved by generating multiple prediction models each with a different feature subset [30].

d) Prediction Cascades: Many real-world problems such as object tracking, speech recognition, and machine translation require performing a sequence of interdependent predictions, forming prediction cascades. Inter-dependencies between the prediction tasks can be typically tackled by stage-wise parallelization of individual tasks, along with adaptive task management. But, this is a non-trivial task.

e) Inference Time Constraints: The inference time can be reduced in machine learning applications by employing concurrent execution of tasks rather than serial. The two major ways of achieving this concurrency are: data parallelism and task parallelism [31]. Data parallelism refers to executing the same computation on multiple inputs concurrently (MapReduce paradigm), whereas task parallelism is achieved when algorithm execution can be partitioned into segments that are independent and hence can be executed concurrently by using parallel multi-core or GPU based implementations.

6. Use Cases

Organizations are increasingly turning to big data to discover new ways to improve decision-making, opportunities and overall performance [32]. There are many use cases of big data in various domains as shown in figure **14** and figure **15**. In this section, we discuss a few use cases of Big Data.

Fig. 14 Use Cases in Various Domains.

Fraud Detection: In the recent years, Banks and Credit card vendors are monitoring one's spending habits on real-time basis. In addition to the transaction records for authorization and approvals, banks and credit card companies are collecting lot more information from location, life style, spending patterns. Credit card companies manage huge volume of data such as individual Social Security number and income, account balances and employment details, and credit history and transaction history. All this put together helps credit card companies to fight fraud in real-time. Big Data architecture provides the scalability to analyze the incoming transactions against

individual history and approve/decline the transaction and alert the account owner. The GE Consumer & Industrial Home Services Division estimated that it saved about $5.1 million in a year by detecting suspect claims [33].

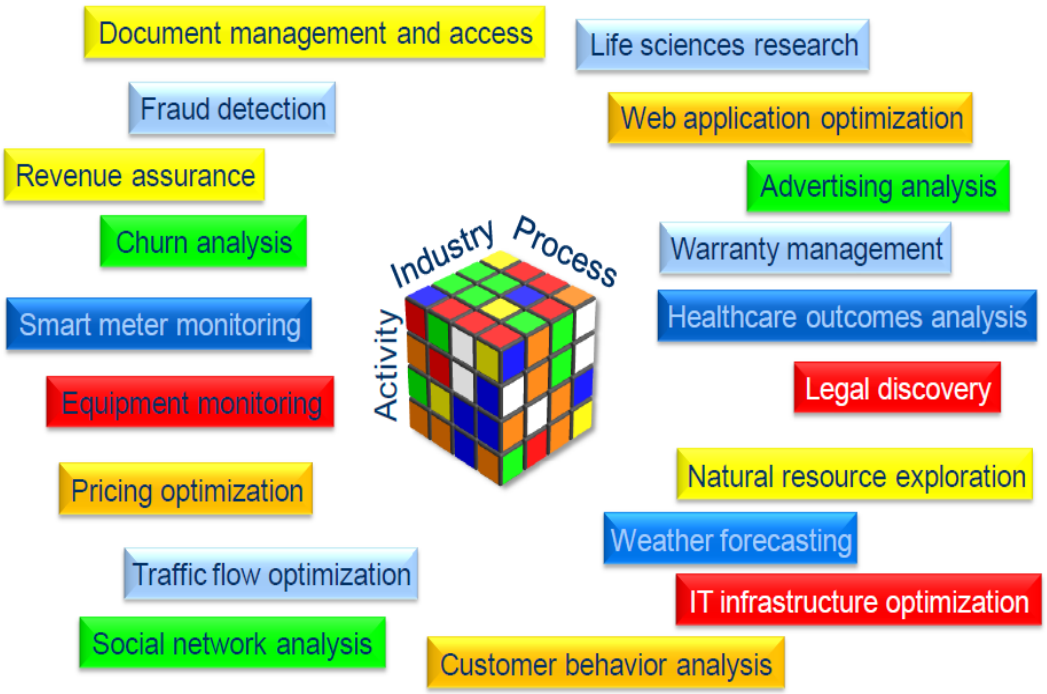

Fig. 15 Use Cases of Big Data (Source: IDC 2012).

Smart Meter/Grids: Evolving technologies in the energy and utilities industry, including smart meters and smart grids, can provide companies with unprecedented capabilities for forecasting demand, shaping customer usage patterns, preventing outages, optimizing unit commitment and more. To manage and use this information to gain insight, utility companies must be capable of high-volume data management and advanced analytics designed to transform data into actionable insights. PG&E was named one of America's most 'intelligent utilities' in the third annual "UtiliQ" ranking by Intelligent Utility magazine and IDC Energy Insights in 2012 [34]. This is because they used BDA to get much better knowledge of consumption patterns, allowing the utility to help customers drive down their costs.

Health Care: A study conducted by McKinsey Global Institute (MGI) for health care predicted that if U.S. health care organizations use big data analytics (BDA) creatively and effectively to drive efficiency and quality, the sector could create more than 300 billion in value annually [35]. BDA technologies that sift through large amounts of data, understand, categorize and learn from it, and then predict outcomes or recommend alternative treatments to clinicians and patients at the point of care. A company called 'Ginger.io' is using the feature-rich smartphones and analytics to monitor the well-being of patients by assessing various metrics of biofeedback, including blood glucose, heart rate and accelerometer monitoring [36].

Customer Segmentation: Marketing is one of the major areas where big data is playing a critical role. Today, customers are more connected to social media & online websites than ever before and there is a lot of competition for their attention. Marketing Research Departments can use segmentation to overcome problems of fragmentation and proliferation to create a holistic and strategic basis for its consumer knowledge base [37]. In addition to the traditional 360-degree view of the customer's external behavior with the world (i.e., their buying, consuming, influencing, churning, and other observable behaviors), extra 360 degrees of internal behavior (i.e, their experiences, propensities, sentiments, attitudes, etc.) culled from behavioral data sources and/or inferred through sophisticated analytics [38]. Ford's Don Butler pointed that Ford's designers and product planners are turning to social media and survey data as they wrestle with product design issues. Ford uses big data to optimize its supply chain and to increase its operational efficiency[39].

Churn Analysis: Now a days customers want competitive pricing, value for money and, above all, a high quality service. They are not hesitating to switch providers if they don't find what they are looking for. The telecom market is a good example for this. Big Data analytics provides an opportunity to telecom business to enable an operator to move from reactive churn management to proactive customer retention based on predictive churn modeling using social media analytics to identify potential 'churners'. T-Mobile USA, with their big data strategy, managed to bring down churn rates by 50% in just one quarter [40].

7. Conclusion

In this paper, our objective is to illustrate the challenges and opportunities of big data in a correlated and standardized way, providing insights on analytics. Hence, In this paper, we have discussed in detail about following points: a) different characteristics i.e. different V's of big data and the challenges associated with these characteristics; b) why the open source data platform Hadoop has become a poster child for big data; c) how NoSQL databases can maneuver gigantic volume of data being generated in myriad formats; d) the challenges due to streaming data and how to address these issues by using tools such as Apache Spark and Apache Storm; e) different data mining and machine learning algorithms used for predictive analytics and the challenges while using these algorithms on big data; f) several use cases of big data finally. Even though, many people having a myth that 'big data' is big hype, we have established here that it is worth considering Hadoop ecosystem technologies to solve some of the real-world problems and use it for taking insightful decisions to revamp the businesses by taking a leap from hindsight to foresight.

8. References

1.The World Economic Forum, 2012, "Big Data, Big Impact: New Possibilities for International Development" , Switzerland .
2.NASSCOM, 2012, "Big Data The Next Big Thing", New Delhi, India.

3.NESSI, 2012, "Big Data: A New World of Opportunities", December.
4.Wei Fan and Albert Bifet, 2014, "Mining Big Data: Current Status, and Forecast to the Future", SIGKDD Explorations, Vol.14(2).
5.http://expandedramblings.com/index.php/by-the-numbers-17-amazing-facebook-stats/8/
6.Leading Researchers, 2012, "Challenges and Opportunities with BigData", Community white paper, United States.
7.Jianqing Fan, Fang Han and Han Liu, 2013, "Challenges of Big Data Analysis", National Science Review, August.
8.Jimmy Lin and Chris Dyer, 2010, "Data-Intensive Text Processing with MapReduce", Morgan & Claypool Synthesis Lectures on Human Language Technologies.
9.Apache MapReduce Tutorial
10.http://nosql-database.org/
11.Dr. S. George, 2013, "NOSQL - NOTONLY SQL", International Journal of Enterprise Computing and Business Systems, Vol. 2(2), July.
12.Justin J. Miller, 2013, "Graph Database Applications and Concepts with Neo4j", Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA March.
13.http://planetcassandra.org/apache-cassandra-use-cases/
14.http://www.mongodb.com/use-cases/internet-of-things
15.http://basho.com/riak-users/
16.Christof Strauch, 2011, "NoSQL Databases", Compiled Lecture Series on selected topics on software-technology: Ultra-Large Scale Sites, Stuttgart Media University.
17.http://blog.beany.co.kr/archives/275
18.Byron Ellis, 2014, "Real-Time Analytics: Techniques to Analyze and Visualize Streaming Data", Wiley, July.
19.https://storm.apache.org/
20.https://spark.apache.org/
21.http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html
22.Dursun Delena, Haluk Demirkanb, 2013, "Data, information and analytics as services", Decision Support Systems, Vol 55(1), pp. 359-363, April.
23.Shih-Ying Chen, Jia-Hong Li, Ke-Chung Lin, Hung-Ming Chen and Tung-Shou Chen, 2013, "Using MapReduce Framework for Mining Association Rules", Lecture Notes in Electrical Engineering, Vol. 253, pp. 723-731.
24.Rainer Gemulla, Peter J. Haas, Erik Nijkamp and Yannis Sismanis, 2013, "Large-Scale Matrix Factorization with Distributed Stochastic Gradient Descent", IBM Technical Report, February.
25.Yehuda Koren, Robert Bell and Chris Volinsky, 2009, "Matrix factorization techniques for recommender systems", IEEE Computer Society.
26.Xuan Liu, 2014, "An Ensemble Method for Large Scale Machine Learning with Hadoop MapReduce", Master's thesis, University of Ottawa, Canada.
27.J Weston., 2008, "Large-scale semi-supervised learning", Proceedings of the

NATO Advanced Study Institute on Mining Massive Data Sets for Security.
28. Mohamed Medhat Gaber, 2012, "Advances in data stream mining", WIREs Data Mining and Knowledge Discovery, Vol. 2.
29. Gaber, Mohamed Medhat, Shonali Krishnaswamy, and Arkady Zaslavsky, 2003, "Adaptive mining techniques for data streams using algorithm output granularity", The Australasian Data Mining Workshop.
30. Yvan Saeys, Thomas Abeel, and Yves Van de Peer, 2008, "Robust Feature Selection Using Ensemble Feature Selection Techniques", ECML PKDD 2008, LNAI 5212, pp. 313-325.
31. Ron Bekkerman, Mikhail Bilenko and John Langford, 2012, "Scaling Up Machine Learning: Parallel and Distributed Approaches", Cambridge University Press.
32. Gautham Vemuganti et al., 2013, "BIG DATA: CHALLENGES AND OPPORTUNITIES", Infosys Labs Briefings, Vol.11(1).
33. Ruchi Verma and Sathyan Ramakrishna Mani, 2012, "Using analytics for insurance fraud detection".
34. Jason Deign, 2013, "DATA MANAGEMENT AND ANALYTICS FOR UTILITIES In-depth Briefing", FC Business Intelligence Ltd.
35. Canada Health Infoway, 2013, "Big Data Analytics in Health".
36. Timothy Schultz, 2013, "Turning Healthcare Challenges into Big Data Opportunities: A Use-Case Review Across the Pharmaceutical Development Lifecycle", Bulletin of the Association for Information Science and Technology, Vol. 39, Number 5, July.
37. Soumendra Mohanty, Madhu Jagadeesh, Harsha Srivatsa, 2013, "Big Data Imperatives: Enterprise 'Big Data' Warehouse, 'BI' Implementations and Analytics", Apress Publishers.
38. James Kobielus, 2012, "Targeted Marketing: When Does Cool Cross Over to Creepy?".
39. https://datafloq.com/read/ford-drives-direction-big-data/434
40. https://datafloq.com/read/t-mobile-usa-cuts-downs-churn-rate-with-big-data/512G. J. Alred, C. T. Brusaw, and W. E. Oliu, *Handbook of Technical Writing*, 7th ed., St. Martin's, New York (2003).