Challenges and opportunities in Big data: A review

Er. Amar Paul Singh[1], Dr. Yogesh Mohan[2]
[1]*Research Scholar, Computer Science Department, Himachal Pradesh University, Shimla*
[2]*Assistant Professor, Computer Science Department, Himachal Pradesh University, Shimla*

Abstract: Big Data promise new levels of scientific discovery and economic value. Big Data bring new opportunities to modern society and challenges to data scientists The Big Data revolution promises to transform how we live, work, and think by enabling process optimization, empowering insight discovery and improving decision-making. The realization of this grand potential relies on the ability to extract value from such massive data through data analytics, machine learning is at its core because of its ability to learn from data and provide data driven insights, decisions, and predictions. However, traditional machine learning approaches were developed in a different era and thus are based upon multiple assumptions, such as the dataset fitting entirely into memory, what unfortunately no longer holds true in this new context. Big data refers to the large volume of complex, (semi) structured, and unstructured data that are generated in a large size and that arrive (in a system) at a higher speed so that it can be analyzed for better decision making and strategic organization and business move

Keywords: Big Data, Data Analysis, Data, data storage, scalability, Data Science,ML, Hadoop.

I. INTRODUCTION

TODAY, the amount of data is exploding at an unprecedented rate as a result of developments in Web technologies, social media, and mobile and sensing devices. For example, Twitter processes over 70M tweets per day, thereby generating over 8TB daily [1]. ABI Research estimates that by 2020, there will be more than 30 billion connected devices [2]. These Big Data possess tremendous potential in terms of business value in a variety of fields such as health care, biology, transportation, online advertising, energy management, and financial services [3], [4]. However, traditional approaches are struggling when faced with these massive data. The concept of Big Data is defined by Gartner [5] as high volume, high velocity, and/or high variety data that require new processing paradigms to enable insight discovery, improved decision making, and process optimization. According to this definition, Big Data are not characterized by specific size metrics, but rather by the fact that traditional approaches are struggling to process them due to their size, velocity or variety. The potential of Big Data is highlighted by their definition; however, realization of this potential depends on improving traditional approaches or developing new ones capable of handling such data. Because of their potential, Big Data have been referred to as a revolution that will transform how we live, work, and think [6]. The main purpose of this revolution is to make use of large amounts of data to enable knowledge discovery and better decision making [6]. The ability to extract value from Big Data depends on data analytics; Jagadish *et al*. [7] consider analytics to be the core of the Big Data revolution. Data analytics[17] involves various approaches, technologies, and tools such as those from text analytics, business intelligence, data visualization, and statistical analysis. This paper focusses on machine learning (ML) as a fundamental component of data analytics. The McKinsey Global Institute has stated that ML will be one of the main drivers of the Big Data revolution [8]. The reason for this is its ability to learn from data and provide data driven insights, decisions, and predictions [9]. It is based on statistics and, similarly to statistical analysis, can extract trends from data; however, it does not require the explicit use of statistical proofs. According to the nature of the available data, the two main categories of learning tasks are: *supervised learning* when both inputs and their desired outputs (labels) are known and the system learns to map inputs to outputs and *unsupervised learning* when desired outputs are not known and the system itself discovers the structure within the data. Classification and regression are examples of supervised learning: in classification the outputs take discrete values (class labels) while in regression the outputs are continuous. Examples of classification algorithms are k-nearest neighbor, logistic regression, and Support Vector Machine (SVM) while regression examples include Support Vector Regression (SVR), linear regression, and polynomial regression. Some algorithms such as neural networks can be used for both, classification and regression. Unsupervised learning includes clustering which groups objects based on established similarity criteria; k-means is an example of such algorithm. Predictive analytics relies on machine learning to develop models built using past data in an attempt to predict the future [10]; numerous algorithms including SVR, neural networks, and Naïve Bayes can be used for this purpose.

II. BACKGROUND

We are entering the era of Big Data—a term that refers to the explosion of available information. Such a Big Data movement is driven by the fact that massive amounts of very high-

dimensional or unstructured data are continuously produced and stored with much cheaper cost than they used to be drop in price for whole genome sequencing [1]. This is also true in other areas such as social media analysis, biomedical imaging, high-frequency finance, analysis of surveillance videos and retail sales. The existing trend that data can be produced and stored more massively and cheaply is likely to maintain or even accelerate in the future [2]. This trend will have deep impact on science, engineering and business. For example, scientific advances are becoming more and more data-driven and researchers will more and more think of themselves as consumers
of data. The massive amounts of high-dimensional data bring both opportunities and new challenges to data analysis. Valid statistical analysis for Big Data is becoming increasingly important.

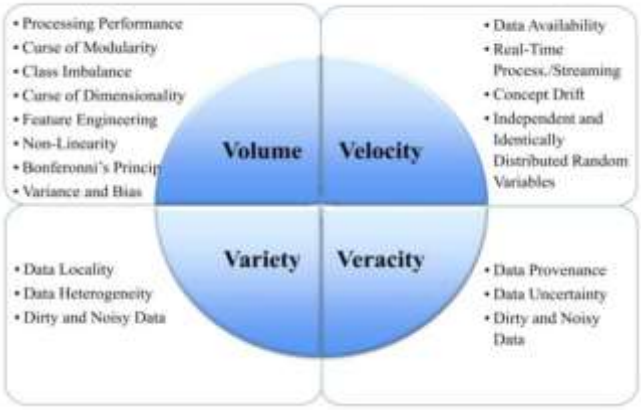

Fig. 1. Big Data characteristics with associated challenges

III. RELATED-WORK

Generally, Big Data is utilized for the management of datasets that are large in size and are beyond the ability of a common software to manage and analyze in an efficient manner. With the advancement in technologies, it is expected that the generation of data will double in the coming years [2]. The authors examined the strategic journey regarding big data privacy protection. The authors have stated that big data can be stored effectively and efficiently with the use of a number of strategies. However, the important thing is to consider the big data privacy lookup [3]. The authors for their research have surveyed the privacy techniques, obstacles, and requirements that are associated with the protection of data. An important role is being played by big data protection as it enables the confidentiality of data. Compared to data privacy, data security is different. Privacy concentrates only on a specific person using the data for making sure that it is being used in the right way. In Big Data analytics, privacy is essential due to a number of reasons. Weak protection techniques prove to be inefficient when it comes to Big Data [4].

IV. GOALS AND CHALLENGES OF ANALYZING BIG DATA

What are the goals of analyzing Big Data? According to [3], two main goals of high-dimensional data analysis are to develop effective methods that can accurately predict the future observations and at the same time to gain insight into the relationship between the features and response for scientific purposes. Furthermore, due to large sample size, Big Data give rise to two additional goals: to understand heterogeneity and commonality across different sub populations. In other words, Big Data give promises for:

(i) exploring the hidden structures of each subpopulation of the data, which is traditionally not feasible and might even be treated as 'outliers' when the sample size is small. (ii) Extracting important common features across many subpopulations even when there are large individual variations. What are the challenges of analyzing Big Data? Big Data are characterized by high dimensionality and large sample size. These two features raise three unique challenges: (i) high dimensionality brings noise accumulation, spurious correlations and incidental homogeneity; (ii) high dimensionality combined with large sample size creates issues such as heavy computational cost and algorithmic instability; (iii) the massive samples in Big Data are 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 .

Fig2. Big data Challenges [11]

A.Volume

The first and the most talked about characteristic of Big Data is volume: it is the amount, size, and scale of the data. In the machine learning context, size can be defined either vertically by the number of records or samples in a dataset or horizontally by the number of features or attributes it contains. Furthermore, volume is relative to the type of data: a smaller number of very complex data points may be considered equivalent to a larger quantity of simple data [11]. This is perhaps the easiest dimension of Big Data to define, but at the same time, it is the cause of numerous challenges. The following sub-sections discuss machine learning challenges caused by volume.

1) Processing Performance :One of the main challenges encountered in computations with Big Data comes from the simple principle that scale, or volume, adds computational complexity. Consequently, as the scale becomes large, even trivial operations can become costly. For example, the standard support vector machine (SVM) algorithm has a training time complexity of O(m3) and a space complexity of O(m2) , where m is the number of training samples. Therefore, an increase in the size m will drastically affect the time and memory needed to train the SVM algorithm and may even become computationally infeasible on very large datasets. Many other ML algorithms also exhibit high time complexity: for example, the time complexity of principal component analysis is O(mn2+n3), that of logistic regression O(mn2+n3), that of locally weighted linear regression O(mn2+n3), and that of Gaussian discriminative analysis O(mn2+n3) , where m is the number of samples and n the number of features. Hence, for all these algorithms, the time needed to perform the computations will increase exponentially with increasing data size and may even render the algorithms unusable for very large datasets. Moreover, as data size increases, the performance of algorithms becomes more dependent upon the architecture used to store and move data. Parallel data structures, data partitioning and placement, and data reuse become more important with growth in data size [15]. Resilient distributed datasets (RDDs) [31] are an example of a new abstraction for in-memory computations on large clusters; RDDs are implemented in the Spark cluster computing framework . Therefore, not only does data size affect performance, but it also leads to the need to re-think the typical architecture used to implement and develop algorithms.

2) Curse of Modularity Many learning algorithms rely on the assumption that the data being processed can be held entirely in memory or in a single file on a disk . Multiple classes of algorithms are designed on strategies and building blocks that depend on the validity of this assumption. However, when data size leads to the failure of this premise, entire families of algorithms are affected. This challenge is referred to as the curse of modularity [15]. One of the approaches brought forward as a solution for this curse is MapReduce, a scalable programming paradigm for processing large datasets by means of parallel execution on a large number of nodes. Some machine learning algorithms are inherently parallel and can be adapted to the MapReduce paradigm, whereas others are difficult to decompose in a way that can take advantage of large numbers of computing nodes.Grolinger et al. [11] have discussed challenges for MapReduce in Big Data. The three main categories of algorithms that encounter the curse of modularity when attempting to use the MapReduce paradigm include iterative graph, gradient descent, and expectation maximization algorithms. Their iterative nature together with their dependence on in-memory data create a disconnect with the parallel and distributed nature of MapReduce. This leads to difficulties in adapting these families of algorithms to MapReduce or to another distributed computation paradigm. Consequently, although some algorithms such as k-means can be adapted to overcome the curse of modularity through parallelization and distributed computing, others are still bounded or even unusable with certain paradigms.

3) Class Imbalance As datasets grow larger, the assumption that the data are uniformly distributed across all classes is often broken .This leads to a challenge referred to as class imbalance: the performance of a machine learning algorithm can be negatively affected when datasets contain data from classes with various probabilities of occurrence. This problem is especially prominent when some classes are represented by a large number of samples and some by very few. Class imbalance is not exclusive to Big Data and has been the subject of research for more than a decade . Experiments performed by Japkowicz and Stephen have shown that the severity of the imbalance problem depends on task complexity, the degree of class imbalance, and the overall size of the training set. They suggest that in large datasets, there is a good chance that classes are represented by a reasonable number of samples; however, to confirm this observation, evaluations of real-world Big Data sets are needed. On the other hand, the complexity of Big Data tasks is expected to be high, which could result in severe impacts from class imbalance. It is to expect that this challenge would be more common, severe, and complex in the Big Data context because the extent of imbalance has immense potential to grow due to increased data size. The same authors, Japkowicz and Stephen , showed that decision trees, neural networks, and support vector machine algorithms are all very sensitive to class imbalance. Therefore, their unaltered execution in the Big Data context without addressing class imbalance may produce inadequate results. Similarly, Baughman et al. considered extreme class imbalance in gamification and demonstrated its negative effects on Watson machine learning. Consequently, in the Big Data context, due to data size, the probability that class

imbalance will occur is high. In addition, because of the complex problems embedded in such data, the potential effects of class imbalance on machine learning are severe**.**

4) Curse of Dimensionality Another issue associated with the volume of Big Data is the curse of dimensionality which refers to difficulties encountered when working in high dimensional space. Specifically, the dimensionality describes the number of features or attributes present in the dataset. The Hughes effect [11] states that for a training set of static size, the predictive ability and effectiveness of an algorithm decreases as the dimensionality increases. Therefore, as the number of features increases, the performance and accuracy of machine learning algorithms degrades. This can be explained by the breakdown of the similarity-based reasoning upon which many machine learning algorithms rely. Unfortunately, the greater the amount of data available to describe a phenomenon, the greater becomes the potential for high dimensionality because there are more prospective features. Consequently, as the volume of Big Data increases, so does the likelihood of high dimensionality. In addition, dimensionality affects processing performance: the time and space complexity of ML algorithms is closely related to data dimensionality. The time complexity of many ML algorithms is polynomial in the number of dimensions. As already mentioned, the time complexity of the principal component analysis is O(mn2+n3) and that of logistic regression O(mn2+n3), where m is the number of samples and n is the number of dimensions**.**

5) Feature Engineering High dimensionality is closely related to another volume challenge: feature engineering. It is the process of creating features, typically using domain knowledge, to make machine learning perform better. Indeed, the selection of the most appropriate features is one of the most time consuming preprocessing tasks in machine learning [15]. As the dataset grows, both vertically and horizontally, it becomes more difficult to create new, highly relevant features. Consequently, in a manner similar to dimensionality, as the size of the dataset increases, so do the difficulties associated with feature engineering. Feature engineering is related to feature selection: whereas feature engineering creates new features in an effort to improve learning outcomes, feature selection (dimensionality reduction) aims to select the most relevant features. Although feature selection reduces dimensionality and hence has the potential to reduce ML time, in high dimensions it is challenging due to spurious correlations and incidental endogeneity (correlation of an explanatory variable with the error term).
Overall, both feature selection and engineering are still very relevant in the Big Data context, but, at the same time they become more complex.

6**) Non-Linearity** Data size poses challenges to the application of common methodologies used to evaluate dataset characteristics and algorithm performance. Indeed, the validity of many metrics and techniques relies upon a set of assumptions, including the very common assumption of linearity. For example, the correlation coefficient is often cited as a good indicator of the strength of the relationship between two or more variables. However, the value of the coefficient is only fully meaningful if a linear relationship exists between these variables. An experiment conducted by Kiang showed that the performance of neural networks and logistic regression is very negatively affected by non-linearity. Although this problem is not exclusive to Big Data, non-linearity can be expected to be more prominent in large datasets. The challenge of non-linearity in Big Data also stems from the difficulties associated with evaluating linearity. Linearity is often evaluated using graphical techniques such as scatterplots; however, in the case of Big Data, the large number of points often creates a large cloud, making it difficult to observe relationships and assess linearity. Therefore, both the difficulty of assessing linearity and the presence of non-linearity pose challenges to the execution of machine learning algorithms in the context of Big Data.

7) Bonferonni's Principle Bonferonni's principle embodies the idea that if one is looking for a specific type of event within a certain amount of data, the likelihood of finding this event is high. However, more often than not, these occurrences are bogus, meaning that they have no cause and are therefore meaningless instances within a dataset. This statistical challenge is also often described as spurious correlation [11]. In statistics, the Bonferonni correction theorem provides a means of avoiding those bogus positive searches within a dataset. It suggests that if testing m hypotheses with a desired significance of α, each individual hypothesis should be tested at a significance level of α/m . However, the incidences of such phenomena increase with data size, and as data become exponentially bigger, the chances of finding an event of interest, legitimate or not, is bound to increase. Recently, Calude and Longo have discussed the impact and incidence of spurious correlations in Big Data. They have shown that given a large enough volume, most correlations tend to be spurious. Therefore, including a means of preventing those false positives is important to consider in the context of machine learning with Big Data.

8) Variance and Bias Machine learning relies upon the idea of generalization; through observations and manipulations of data, representations can be generalized to enable analysis and prediction. Generalization error can be broken down into two components: variance and bias. Variance describes the consistency of a learner's ability to predict random things, whereas bias describes the ability of a learner to learn the wrong thing. Ideally, both the variance and the bias error should be minimized to obtain an accurate output. However, as the volume of data increases, the learner may become too closely

biased to the training set and may be unable to generalize adequately for new data. Therefore, when dealing with Big Data, caution should be taken as bias can be introduced, Regularization refers to techniques that aim to improve generalization and reduce overfitting; examples of regularization techniques include early stopping, Lasso, and Ridge . Although these techniques improve generalization, they also introduce additional parameters that must be tuned to achieve good fit to unseen data. This is often done using approaches such as cross-validation, possibly with grid search; however, those require additional processing time, especially in the case of large datasets. Regularization techniques are well established in machine learning, but further investigation is needed with respect to their efficiency with Big Data.

B. Variety The ***variety*** of Big Data describes not only the structural variation of a dataset and of the data types that it contains, but also the variety in what it represents, its semantic interpretation [7] and its sources. Although not as many as for other V dimensions, the challenges associated with this dimension have substantial impact**. 1) Data Locality**. The first challenge associated with variety is data locality. Machine learning algorithms once again assume that the entire dataset is found in memory or in a single disk file [15]. However, in the case of Big Data, this may not be possible due to sheer size; not only do the data not fit into memory, but they are commonly distributed over large numbers of files residing in different physical locations. Traditional machine learning would first require data transfer to the computing location. With large datasets, transfer would result in processing latency and could cause massive network traffic. Consequently, an approach of bringing computation to data as opposed to bringing data to computation has emerged. This is based on the premise that moving computation is cheaper, in terms of time and bandwidth, than moving data. This approach is especially prominent with Big Data. The Map Reduce paradigm also uses it: map tasks are executed on the nodes where data reside, with each map task processing its local data. Moreover, a large number of NoSQL data stores adapt this model; as distributed storage solutions, they store data over a large number of nodes and then use the Map Reduce paradigm to bring computation to data. However, as already mentioned, Map Reduce-based approaches encounter difficulties when working with highly iterative algorithms. With small datasets, physical location is a non-issue; however, with Big Data, data locality is a paramount challenge that must be addressed in any successful Big Data system.

2) Data Heterogeneity Big Data analytics often involve integrating diverse data from several sources. These data may be diverse in terms of data type, format, data model, and semantics. Two main heterogeneity categories can be recognized: syntactic and semantic heterogeneity. Syntactic heterogeneity refers to diversity in data types, file formats, data encoding, data model, and similar. To carry out analytics with integrated datasets, these syntactic variations must be reconciled [7]. Machine learning often requires a data pre-processing and cleaning step to configure data to fit within a specific model. However, with data coming from different sources, these data are likely formatted differently. Furthermore, the data to be processed may be of completely different type

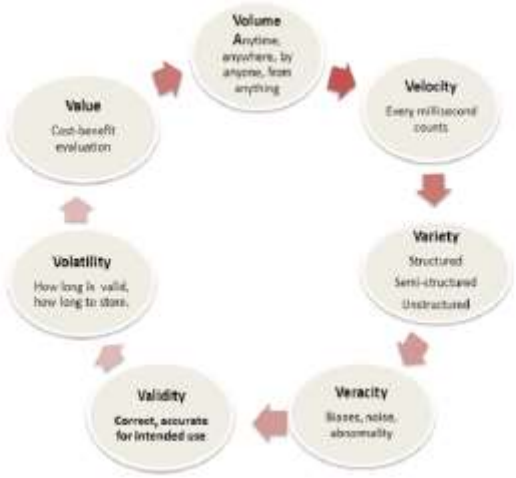

Fig3: Characteristics of Big Data.

for example, images may need to be processed along with categorical and numerical data. This causes difficulties for machine learning algorithms because they are not designed to recognize various types of representations at one time and to create efficient unified generalizations. Semantic heterogeneity refers to differences in meanings and interpretations. As with syntactic, semantic heterogeneity increases in the case of Big Data when a number of datasets developed by different parties are integrated . Again, machine learning approaches were not developed to handle semantically diverse data, and therefore heterogeneity must be resolved before applying such approaches. In statistics, heterogeneity also refers to differences in statistical properties among the different parts of an overall dataset. Although present in small datasets, this challenge is enlarged in Big Data because datasets typically involve parts coming from different sources. This statistical heterogeneity breaks the common machine learning assumption that statistical properties are similar across a complete dataset. Both syntactic and semantic heterogeneity as well as statistical heterogeneity have been active research topics for a long time, but with the emergence of Big Data, they have attracted renewed attention . The business value of data analytics typically involves correlating diverse datasets, and

integration is crucial for carrying out machine learning over such datasets.

3)Dirty and Noisy Data

According to Ratner , data possess their own set of distinct features that can be used for characterization:
• Condition defines the readiness of the data for analysis.
• Location refers to where the data physically reside.
• Population describes the entities and their sets of common attributes that together form the dataset.
Big Data are typically described as ill-conditioned due to the amount of time and resources necessary to get them ready for analysis. They also come from various locations and unknown populations. The combination of these properties leads to Big Data often being described as dirty.
Fan et al. referred to such data as noisy data; they contain various types of measurement errors, outliers, and missing values. They discussed noise accumulation, which is especially severe with the high dimensionality typical in Big Data. It is important to note that Fan et al. considered noisy data one of the three main challenges of Big Data analysis.
Swan suggested that data analysis should include a step to extract signal from noise directly following the steps of data collection and integration. She also recognized that Big Data may be too noisy to produce meaningful results.
The studies described above demonstrate the importance of dealing with noise in the context of generic Big Data analysis. Likewise, noise needs to be considered in machine learning withBigData.

C.Velocity

The velocity dimension of Big Data refers not only to the speed at which data are generated, but also the rate at which they must be analyzed. With the omnipresence of smartphones and real-time sensors and the impending need to interact quickly with our environment through the development of technologies such as smart homes, the velocity of Big Data has become an important factor to consider.

1)Data Availability

Historically, many machine learning approaches have depended on data availability, meaning that before learning began, the entire dataset was assumed to be present. However, in the context of streaming data, where new data are constantly arriving, such a requirement cannot be fulfilled. Moreover, even data arriving at non-real-time intervals may pose a challenge.
In machine learning, a model typically learns from the training set and then performs the learned task, for example classification or prediction, on new data. In this scenario, the model does not automatically learn from newly arriving data, but instead carries out the already learned task on new data. To accommodate the knowledge embedded in new data, these models must be retrained. Without retraining, they may become outdated and cease to reflect the current state of the system. Therefore, to adapt to new information, algorithms must support incremental learning, sometimes referred to as sequential learning, which is defined as an algorithm's ability to adapt its learning based on the arrival of new data without the need to retrain on the complete dataset. This approach does not assume that the entire training set is available before learning begins, but processes new data as they arrive. Although incremental learning is a relatively old concept, it is still an active research area due to the difficulty of adapting some algorithms to continuously arriving data.

2)Real-Time Processing/Streaming

Similar to the already discussed data availability challenge, traditional machine learning approaches are not designed to handle constant streams of data [19], which leads to another velocity dimension challenge - the need for real-time processing. This is subtly different from the data availability challenge: whereas data availability refers to the need to update the ML model as new data arrive, real-time processing refers to the need for real-time or near-real-time processing of fast-arriving data. The business value of real-time processing systems lies in their ability to provide instantaneous reaction; developers of algorithmic trading, fraud detection, and surveillance systems have been especially interested in such solutions [11].
The importance of real-time processing in today's era of sensors, mobile devices, and IoT has resulted in the emergence of a number of streaming systems; examples include Twitter's Storm and Yahoo's S4 . Although those systems have seen great success in real-time processing, they do not include sophisticated or diverse ML, but users can add ML features using external languages or tools.
The need exists to merge these streaming solutions with machine learning algorithms to provide instantaneous results; however, the complexity of such algorithms and the sparse availability of online learning solutions make this a difficult task.

3)ConceptDrift

Big Data are non-stationary; new data are arriving continuously. Consequently, acquiring the entire dataset before processing it is not possible, meaning that it cannot be determined whether the current data follow the same distribution as future data. This leads to another interesting challenge in machine learning with Big Data: concept drift [15]. Concept drift can be formally defined as changes in the conditional distribution of the target output given the input, while the distribution of the input itself may remain unchanged . Specifically, this leads to a problem that occurs when machine learning models are built using older

data that no longer accurately reflect the distribution of new data . For example, energy consumption and demand prediction models can be built using data from electricity meters, but when buildings are retrofitted to improve their energy efficiency, the present model does not accurately represent the new energy characteristics. Sliding window is a possible way of dealing with concept drift: the model is built using only the samples from the training window which is moved to include only the most recent samples. Windowing approach assumes that the most recent data is more relevant which may not always be true .
There exist various types of concept drift: incremental, gradual, sudden, and recurring each bringing its own set of issues. However, the challenges typically lie in quickly detecting when concept drift is occurring and effectively handling the model transition during these changes. Like several already mentioned concepts, concept drift is not a new issue; mentions of it date back to 1986 . However, the advent and nature of Big Data have increased frequency of its occurrence and have rendered some previous methodologies unusable. For example, Lavaire et al. conducted an experiment on the influence of high dimensional Big Data on existing concept drift mitigation techniques. Their conclusions were that algorithm performance was highly degraded by the changes in the data. Therefore, finding new means to handle concept drift in the context of Big Data is an important task for the future of machine learning.

4) Independent and Identically Distributed Random Variables

Another common assumption in machine learning is that random variables are independent and identically distributed (i.i.d.) [19]; it simplifies underlying methods and improves convergence. In other words, i.i.d. assumes that each random variable has the same probability distribution as the others and that all are mutually independent. In reality, this may or may not be true. Moreover, some algorithms also depend on other distributions; for example the Markov sequence assumes that probability distribution of the next state depends only on the current state
Nonetheless, Big Data by their very nature may prevent reliance on i.i.d. assumption based on the following [15]:
• i.i.d. requires data to be in random order while many datasets have a pre-existing non-random order. A typical solution would be to randomize the data before applying the algorithms. However, when dealing with Big Data, this becomes a challenge of its own and is often impractical.
• By their very nature, Big Data are fast and continuous. It is therefore not realistic to randomize a dataset that is still incomplete, nor is it possible to wait for all the data to arrive.Dundar et al. have shown that many typical machine learning algorithms such as back-propagation neural networks and support vector machines depend upon this assumption and could benefit greatly from a way of accounting forit.The high likelihood of a broken i.i.d. assumption with Big Data makes this challenge an important one to address.

D.Veracity

The veracity of Big Data refers not only to the reliability of the data forming a dataset, but also, as IBM has described, to the inherent unreliability of data sources [19]. The provenance and quality of Big Data together define the veracity component , but also pose a number of challenges as discussed in the following sub-sections.

1)DataProvenance

Data provenance is the process of tracing and recording the origin of data and their movements between locations . Recorded information, the provenance data, can be used to identify the source of processing error since it identifies all steps, transactions, and processes undergone by invalid data, thus providing contextual information to machine learning. It is therefore important to capture and retain this metadata [7]. However, as pointed out by Wang et al. , in the context of Big Data, the provenance dataset itself becomes too large, therefore, while these data provide excellent context to machine learning, the volume of these metadata creates its own set of challenges. Moreover, not only is this dataset too large, but the computational cost of carrying this overhead becomes overwhelming . Although, certain methods have been brought forward to capture data provenance for specific data processing paradigms, such as the Reduce and Map Provenance (RAMP) developed for MapReduce as an extension for Hadoop , the added burden of provenance generally adds to the already high complexity and computational cost of machine learning with Big Data. Consequently, as provenance data provide a way to establish the veracity of Big Data, means of balancing its computational overhead and cost with the veracity value are needed.

2)DataUncertainty

Data are now being gathered about various aspects of our lives in different ways; however, the means and methods used to gather data can introduce uncertainty and therefore impact theveracityofadataset.For example, sentiment data are being collected through social media but although these data are highly important because they contain precious insights into subjective information, the data themselves are imprecise. The certainty and accuracy of this type of data is not objective because it
relies only upon human judgment [20]. The lack of objectivity, or of absolute truth, within the data makes it difficult for a machine learning algorithm to learn from it.
Another recent method of capturing data is crowdsourcing; it solicits services or ideas from a large group of people. The data obtained from crowdsourcing, more particularly those

gathered through participatory sensing, contain an even higher degree of uncertainty than sentiment data [7]. Moreover, inherent uncertainties exist in various types of data, such as weather or economic data for example, and even the most sophisticated data pre-processing methods cannot expunge this intrinsic unpredictability. Once again, machine learning algorithms are not designed to handle this kind of imprecise data, thus resulting in another set of unique challenges for machine learning with Big Data.

3)Dirty and Noisy Data

Furthermore, in addition to being imprecise, data can also be noisy . For example, the labels or contextual information associated with the data may be inaccurate, or readings could be spurious. From the machine learning perspective this is different from imprecise data; having an unclear picture is different from having the wrong picture, although it may yield similar results. However, the noise challenge associated with crowdsourcing has yet to be discussed. Crowdsourcing leads to uncertainty, especially when used for participatory sensing, but it can also lead to noisy data because it makes use of human judgment to assign labels to data. Moreover, the incorrect label can be either purposely or not only influences data veracity, but can also affect the performance of machine learning by potentially providing them with improperly labelled data. Dirty and noisy data are not unique to Big Data, but the means by which they can be handled may not be easily adaptable to large datasets.

Opportunities in Big Data:

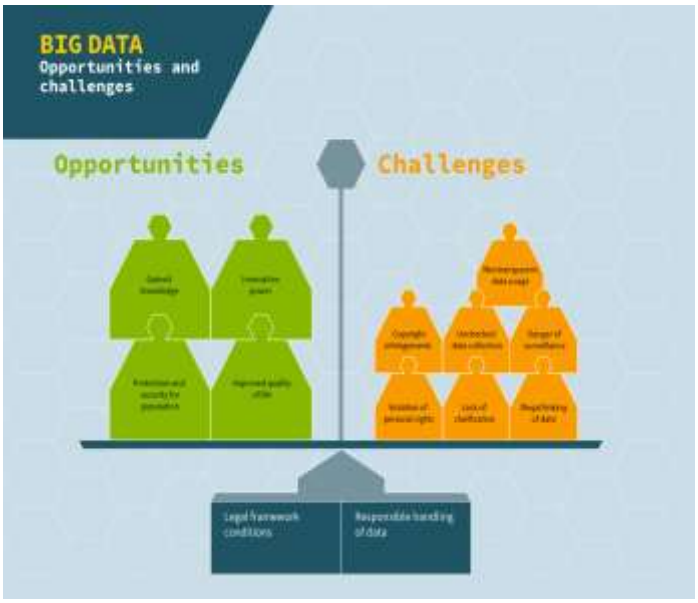

Fig4: Opportunities in Big Data [12]

V. VARIOUS FRAMEWORKS USED FOR BIG DATA

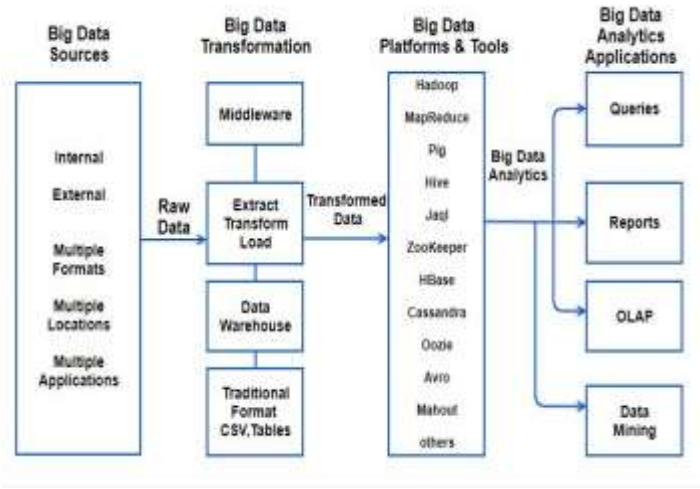

Fig5: General Framework of Big Data Analytics [13]

VI. BIG DATA ANALYTICS FRAMEWORKS

Different types of data when we consider Big Data. Different types of framework required to run different types of analytics. A variety of workloads present in large-scale data processing enterprise. In order to achieve a business goal, we often see a combination of said workloads deployed:

- Batch-oriented processing, for example, Map Reduce based frameworks like Hadoop, for recurring tasks such as large-scale data mining or aggregation [8].

- OLTP, such as user-facing e-commerce transactions, with Apache HBase [14]

- Stream processing, to handle stream sources such as social media feeds or sensor data, with Storm being a representative framework [9].

- Interactive ad-hoc query and analysis with Apache Drill [5].

A. **Apache Hadoop** Apache Hadoop is open source software library which includes framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It has a variety of options ranging from single computer to thousands of computers, each of which offering local computation and storage. Instead of depending on hardware, library itself designed to detect and handle failure and assure high-availability at application layer [7]. Apache Hadoop include following modules:

a) Hadoop core: Common utilities that support other modules.

b) Hadoop distributed file system: Provide high throughput access to application data.

c) Hadoop YARN: Framework for job scheduling and resource management.

d) Hadoop Map Reduce: Framework for parallel processing of large data set.

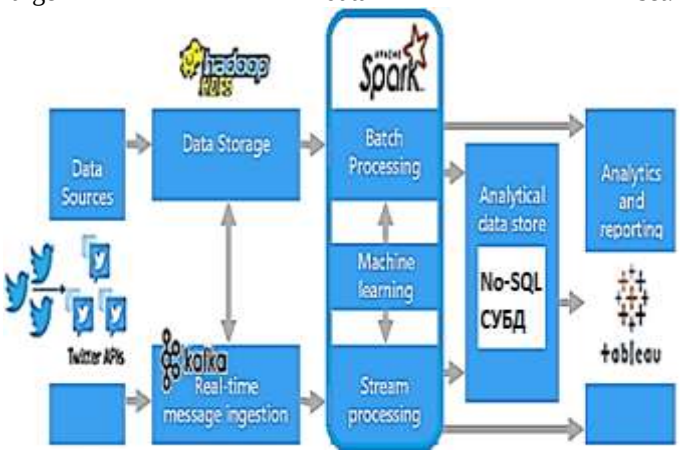

Fig. 6. Data store and retrieval in Apache Hadoop system [14]

Here query is submitted by user to Hadoop Engine which will take input data from HDFS. Data is spread across number of data nodes. There is one name node or Job Tracker which will take care of assigning the work among data nodes and producing the result and responding back to user. Architecture of Apache Hadoop is very robust and fault-tolerant. Job Tracker is continuously tracing the status of data node and if data node remains silent for more than predefined time, task of that data node is given to another data node. B. Project Storm Hadoop and related technologies have made it possible to store and process data at scales previously unthinkable. Unfortunately, these data processing technologies are not real-time systems. However, real-time data processing at massive scale is becoming more and more of a requirement for businesses. Storm exposes a set of primitives for doing real-time computation. Like how Map Reduce greatly eases the writing of parallel batch processing, Storm's primitives greatly ease the writing of parallel real-time computation.

B. Apache Drill [24] Apache Drill is a distributed system for interactive ad-hoc analysis of large-scale datasets. Designed to handle up to petabytes of data spread across thousands of servers, the goal of Drill is to respond to ad-hoc queries in a low latency manner. Many a times it happens that human sits in front of business application and need to execute ad-hoc queries as per business needs. Query should not need more then few seconds to execute even at scale; some time user do not know which query to fire in advance; also, user need to react to changing circumstances. Apache drill will provide the solution for all above issues. At high level Apache Drill's architecture contains following layers: User - providing interfaces such as a command line interface (CLI), a REST interface, JDBC/ODBC, etc., for human or application driven interaction. Processing - allowing for pluggable query languages as well as the query planner, execution, and storage engines. Data sources - pluggable data sources either local or in a cluster setup, providing in-situ data processing. Apache Drill is not a database but rather a query layer that works with a number of underlying data sources. It is primarily designed to do full table scans of relevant data as opposed to, say, maintaining indices. Apache Drill provides for a flexible query execution framework, enabling a number of use cases from quick aggregation of statistics to explorative data analysis. The workers in Apache Drill, suitably called drill-bits, run on each processing node in order to maximize data locality. The coordination of the drill-bits, the query planning, as well as the optimization, scheduling, and execution are performed and distributed

VII. CONCLUSION

In this work a detailed study of Big Data, opportunities and challenges has been performed and comparison between different frameworks is given below:

Features	Apache Hadoop	Project Storm	Apache Drill
Owner	Community	Community	Community
Workload	Batch processing	Real time computation / stream analysis	Interactive and Ad-hoc analysis
Source code	Open	Open	Open
Complexity	Easy	Easy	Complex

TABLE 1 COMPARISION BETWEEN BIG DATA ANALYTICS FRAMEWORKS

As shown in above table, Apache Hadoop is suited for workload where time is not critical factor whereas Project storm is well suited for data stream analysis in which analysis performed is real time and Apache drill is best for interactive and ad-hoc analysis. Following points related to Big Data and Analytics are worth noted. • There is a requirement of Big Data Analytics frameworks for the organization that deal with different types of Big Data workloads. In addition, a middleware architecture is also required to integrate and process all Big Data related workloads. • Organization dealing with Big Data and Analytics need to deal with challenges like privacy, security, data management and sharing, technology, skills and other specific challenges related to workload present in the organization.

VIII. REFERENCES

[1] R. Krikorian, "Twitter by the Numbers," Twitter, 2010. [Online].Available:http://www.slideshare.net/raffikrikorian/twitter-by henumbers?ref=http://techcrunch.com/2010/09/17/twitter-seeing-6-billionapi-calls-per-day-70k-per-second/.

[2] ABI, "Billion Devices Will Wirelessly Connect to the Internet of Everything in 2020," ABI Research, 2013. [Online]. Available:https://www.abiresearch.com/press/more-than-30-billion-devices-willwirelessly-conne/.

[3] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential," Health Information Science and Systems, vol. 2, no. 1, pp. 1–10, 2014.

[4] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K.Taha, "Efficient Machine Learning for Big Data: A Review," Big Data Research, vol. 2, no. 3, pp. 87–93, Apr. 2015.

[5] M. A. Beyer and D. Laney, "The Importance of 'Big Data': a Definition," Gartner Research Report, 2012.

[6] V. Mayer-Schönberger and K. Cukier, Big Data: A Revolution that Will Transform how We Live, Work, and Think. Houghton Mifflin Harcourt,2013.

[7] H. V Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M.Patel, R. Ramakrishnan, and C. Shahabi, "Big Data and its Technical Challenges," Communications of the ACM, vol. 57, no. 7, pp. 86–94,2014.

[8] M. James, C. Michael, B. Brad, and B. Jacques, "Big Data: The Next Frontier for Innovation, Competition, and Productivity," The McKinsey Global Institute, 2011.

[9] M. Rouse, "Machine Learning Definition," 2011. [Online]. Available:http://whatis.techtarget.com/definition/machine-learning.

[10] M. Rouse, "Predictive Analytics Definition," 2009. [Online]. Available:http://searchcrm.techtarget.com/definition/predictive-analytics.

[11] [2] Sam Madden, " From Databases to Big Data", IEEE, Internet Computing, May-June 2012

[12] https://www.google.com/search?q=GOALS+AND+CHALLENGES+OF+ANALYZING+BIG+DATA&tbm=isch&ved=2ahUKEwjWk8bh8tz6AhUWj9gFHWQIAEkQ2CegQIABAA&oq=GOALS+AND+CHALLENGES+OF+ANALYZING+BIG+DATA&gs_lcp=CgNpbWcQA1DDEFjDEGCyKmgAcAB4AIABiAGIAYECkgEDMC4ymAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=69pHY5byFJae4t4P5JCAyAQ&bih=600&biw=1366&rlz=1C1RXQR_enIN1007IN1007#imgrc=E64pRl1PLXRL0M&imgdii=60Tnb-x3d2uWNM

[13] https://medienportal.siemens-stiftung.org/en/big-data-opportunities-and-challenges-112168

[14]https://www.researchgate.net/figure/General-Framework-of-Big-Data-Analytics-8_fig1_319066504

[15] Storing and querying data Big Data in HDFS - http://ecomcanada.wordpress.com/2012/11/14/storing-and-querying-bigdata-in-hadoop-hdfs/

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

[17] Stephen K, Frank A, J. Alberto E, William M, "Big Data: Issues and Challenges Moving Forward", IEEE, 46th Hawaii International Conference on System Sciences, 2013.

[18] Sachchidanand S, Nirmala S, "Big Data Analytics", IEEE, International Conference on Communication, Information & Computing Technology (ICCICT),Oct. 19-20, 2012.

[19] Katina Michael, Keith W. Miller, "Big Data: New Opportunities and New Challenges", IEEE Technology and Society Magazine, vol 13.

[20] Michael Hausenblas, Jacques Nadeau, "Apache Drill Ad-hoc interactive analysis at scale", June 2013. [21] Sergey M, Andrey G, Jing Jing L, Geoffrey R, Shiva S, Matt T, Theo V, "Dremel: Interactive Analysis of Web-Scale Datasets", Google 2013.

[22] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004. [23] "Apache-Hadoop"-http://hadoop.apache.org/#What+Is+Apache+Hadoop%3F

[24] "Project Storm" - http://storm-project.net/

[25] Apache-Drill"-https://cwiki.apache.org/confluence/display/DRILL/Apache+Drill+Wiki

[26] "Structured Data" - http://www.webopedia.com/TERM/S/structured_data.html

[27] Apache HBase - http://hbase.apache.org/

[28] Storing and querying data Big Data in HDFS - http://ecomcanada.wordpress.com/2012/11/14/storing-and-querying-bigdata-in-hadoop-hdfs/

[29] Storm cluster - https://github.com/nathanmarz/storm/wiki/Tutorial [17] Apache Zookeeper - http://zookeeper.apache.org/

[30] Big Data statistics - wikibon.org/blog/big-data-statistics/