OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

A Survey on Big Data Analytics in Data Mining

R. Sathya[1], P. Gouthami[2], C. Sathya[3]

[1, 2.] *(Department of Information Technology, Sri Shakthi Institute of Engineering and Technology, India)*
[3.] *(Department of Computer Science and Engineering, PSNA College of Engineering and Technology, India)*

ABSTRACT: *Big Data is a new term used to identify the datasets that due to their large size, we cannot manage them with the typical data mining software tools. Instead of defining "Big Data" as datasets of a concrete large size, for example in the order of magnitude of petabytes, the definition is related to the fact that the dataset is too big to be managed without using new algorithms or technologies. Big Data analytics – the process of analyzing and mining Big Data – can produce operational and business knowledge at an unprecedented scale and specificity. The need to analyze and leverage trend data collected by businesses is one of the main drivers for Big Data analysis tools. The technological advances in storage, processing, and analysis of Big Data include (a) the rapidly decreasing cost of storage and CPU power in recent years; (b) the flexibility and cost-effectiveness of datacenters and cloud computing for elastic computation and storage; and (c) the development of new frameworks such as Hadoop, which allow users to take advantage of these distributed computing systems storing large quantities of data through flexible parallel processing.*

Keywords: *Big Data, Hadoop, R, Map Reduce*

I. Introduction

Data Mining is an analytic process designed to explore data (usually large amounts of data - typically business or market related - also known as "big data") in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. The ultimate goal of data mining is prediction - and predictive data mining is the most common type of data mining and one that has the most direct business applications. The process of data mining consists of three stages: (1) the initial exploration, (2) model building or pattern identification with validation/verification, and (3) deployment (i.e., the application of the model to new data in order to generate predictions).

Big data analytics refers to the process of collecting, organizing and analyzing large sets of data ("big data") to discover patterns and other useful information. Not only will big data analytics help you to understand the information contained within the data, but it will also help identify the data that is most important to the business and future business decisions. Big data analysts basically want the knowledge that comes from analyzing the data[1].

Big data can also be defined as "Big data is a large volume unstructured data which cannot be handled by standard database management systems like DBMS, RDBMS or ORDBMS".

The primary goal of big data analytics is to help companies make more informed business decisions by enabling data scientists, predictive modelers and other analytics professionals to analyze large volumes of transaction data, as well as other forms of data that may be untapped by conventional business intelligence (BI) programs. That could include Web server logs and Internet click_stream data, social media content and social network activity reports, text from customer emails and survey responses, mobile-phone call detail records and machine data captured by sensors connected to the Internet of Things. Some people exclusively associate big data with semi-structured and unstructured data of that sort, but consulting firms like Gartner Inc. and Forrester Research Inc. also consider transactions and other structured data to be valid components of big data analytics applications.

Big data can be analyzed with the software tools commonly used as part of advanced analytics disciplines such as predictive analytics, data mining, text analytics and statistical analysis. Mainstream BI software and data visualization tools can also play a role in the analysis process. But the semi-structured and unstructured data may not fit well in traditional data warehouses based on relational databases. Furthermore, data warehouses may not be able to handle the processing demands posed by sets of big data that need to be updated frequently or even continually -- for example, real-time data on the performance of mobile applications or of oil and gas pipelines. As a result, many organizations looking to collect, process and analyze big data have turned to a newer class of technologies that includes Hadoop and related tools such as YARN, Map Reduce, Spark, Hive

and Pigas well as NoSQL databases. Those technologies form the core of an open source software framework that supports the processing of large and diverse data sets across clustered systems.

II. Challenges of Big Data Analytics

For most organizations, big data analysis is a challenge. Consider the sheer volume of data and the many different formats of the data (both structured and unstructured data) collected across the entire organization and the many different ways different types of data can be combined, contrasted and analyzed to find patterns and other useful information.

The first challenge is in breaking down data silos to access all data an organization stores in different places and often in different systems. A second big data challenge is in creating platforms that can pull in unstructured data as easily as structured data. This massive volume of data is typically so large that it's difficult to process using traditional database and software methods.

III. Characteristics of Big Data

Big data can be described by the following characteristics[2]:

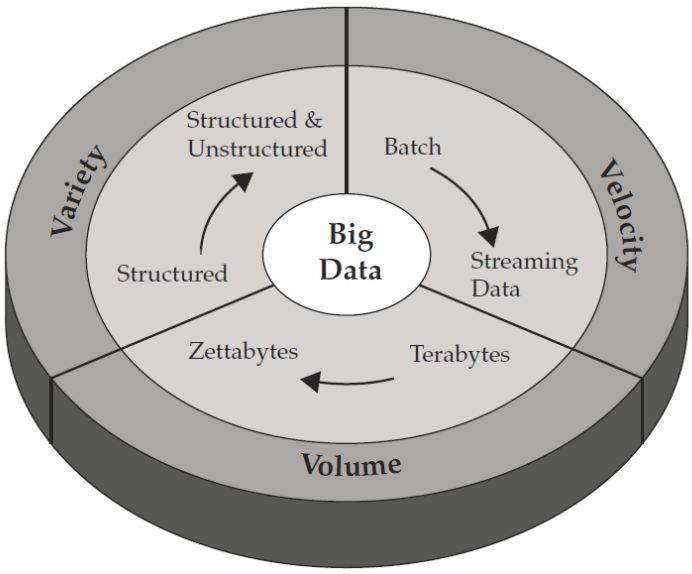
Volume –It is the size of the data which determines the value and potential of the data under consideration and whether it can actually be considered as Big Data or not.

Variety - Means that the category to which Big Data belongs to is also a very essential fact that needs to be known by the data analysts. Helps the people, who are closely analyzing the data and are associated with it, to effectively use the data to their advantage and thus upholding the importance of the Big Data.

Velocity - Refers to the speed of generation of data or how fast the data is generated and processed to meet the demands and the challenges which lie ahead in the path of growth and development.

Variability - A factor which can be a problem for those who analyze the data. Refers to the inconsistency which can be shown by the data at times, thus hampering the process of being able to handle and manage the data effectively.

Complexity - Data management can become a very complex process, especially when large volumes of data come from multiple sources. These data need to be linked, connected and correlated in order to be able to grasp the information that is supposed to be conveyed by these data.

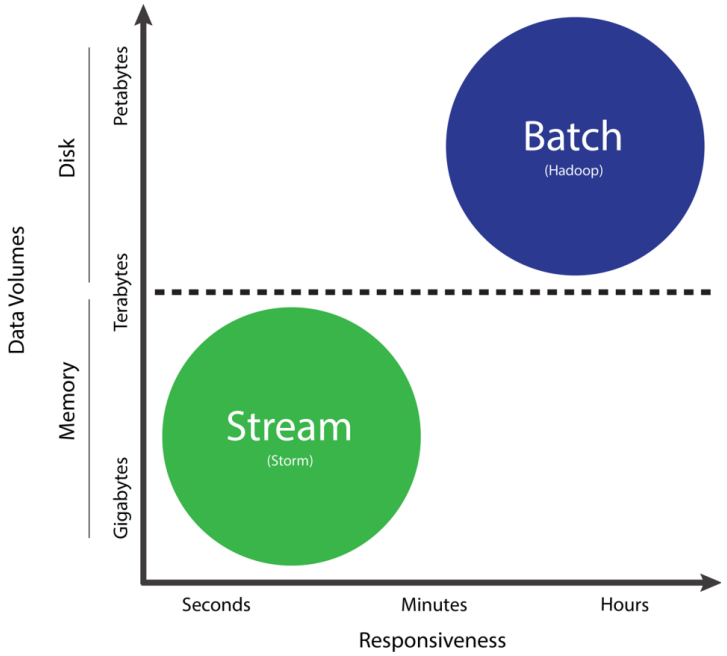


IV. Big Data Technologies

Big Data technologies can be divided into two groups: batch processing, which are analytics on data at rest, and stream processing, which are analytics on data in motion. Real-time processing does not always need to reside in memory, and new interactive analyses of large-scale data sets through new technologies like Drill and Dremel provide new paradigms for data analysis; however, Figure 1 still represents the general trend of these technologies[3].

Hadoop is one of the most popular technologies for batch processing. The Hadoop framework provides developers with the Hadoop Distributed File System for storing large files and the MapReduce programming

model, which is tailored for frequently occurring large-scale data processing problems that can be distributed and parallelized.



V. Big Data Analytics for Security

Big Data analytics can be leveraged to improve information security and situational awareness. For example, Big Data analytics can be employed to analyze financial transactions, log files, and network traffic to identify anomalies and suspicious activities, and to correlate multiple sources of information into a coherent view. Data-driven information security dates back to bank fraud detection and anomaly-based intrusion detection systems. Fraud detection is one of the most visible uses for Big Data analytics. Credit card companies have conducted fraud detection for decades. However, the custom-built infrastructure to mine Big Data for fraud detection was not economical to adapt for other fraud detection uses. Off-the-shelf Big Data tools and techniques are now bringing attention to analytics for fraud detection in healthcare, insurance, and other fields.
In the context of data analytics for intrusion detection, the following evolution is anticipated:

1st generation: Intrusion detection systems – Security architects realized the need for layered security (e.g., reactive security and breach response) because a system with 100% protective security is impossible.

2nd generation: Security information and event management (SIEM) – Managing alerts from different intrusion detection sensors and rules was a big challenge in enterprise settings. SIEM systems aggregate and filter alarms from many sources and present actionable information to security analysts.

3rd generation: Big Data analytics in security (2nd generation SIEM) – Big Data tools have the potential to provide a significant advance in actionable security intelligence by reducing the time for correlating, consolidating, and contextualizing diverse security event information, and also for correlating long-term historical data for forensic purposes.
Analyzing logs, network packets, and system events for forensics and intrusion detection has traditionally been a significant problem; however, traditional technologies fail to provide the tools to support long-term, large-scale analytics for several reasons:

1. Storing and retaining a large quantity of data was not economically feasible. As a result, most event logs and other recorded computer activity were deleted after a fixed retention period (e.g., 60 days).
2. Performing analytics and complex queries on large, structured data sets was inefficient because traditional tools did not leverage Big Data technologies.
3. Traditional tools were not designed to analyze and manage unstructured data. As a result, traditional tools had rigid, defined schemas. Big Data tools (e.g., Piglatin scripts and regular expressions) can query data in flexible formats.
4. Big Data systems use cluster computing infrastructures. As a result, the systems are more reliable and available, and provide guarantees that queries on the systems are processed to completion.

New Big Data technologies, such as databases related to the Hadoop ecosystem and stream processing, are enabling the storage and analysis of large heterogeneous data sets at an unprecedented scale and speed. These technologies will transform security analytics by: (a) collecting data at a massive scale from many internal enterprise sources and external sources such as vulnerability databases; (b) performing deeper analytics on the data; (c) providing a consolidated view of security-related information; and (d) achieving real-time analysis of streaming data. It is important to note that Big Data tools still require system architects and analysts to have a deep knowledge of their system in order to properly configure the Big Data analysis tools.

VI. 'Big Data' Analytics With R and Hadoop

Revolution Analytics addresses both of these opportunities in Big Analytics while supporting the following objectives for working with Big Data Analytics [5]:

- Avoid sampling / aggregation;
- Reduce data movement and replication;
- Bring the analytics as close as possible to the data and;
- Optimize computation speed.

First, Revolution Analytics delivers optimized statistical algorithms for the three primary data management paradigms being employed to address growing size and increasing variety of organizations' data, including file-based, MapReduce (e.g. Hadoop) or In-Database Analytics.

Second, the company is optimizing algorithms - even complex ones - to work well with Big Data. Open Source R was not built for Big Data Analytics because it is memory-bound.

6.1 Revolution Analytics' Capabilities For Hadoop

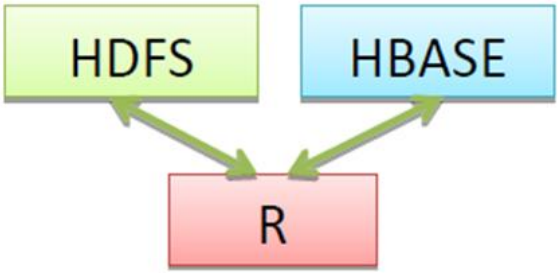
Revolution has created a series of "*RevoConnectRs for Hadoop*" that will allow an R programmer to manipulate Hadoop data stores directly from HDFS and HBASE, and give R programmers the ability to write MapReduce jobs in R using Hadoop Streaming. RevoHDFS provides connectivity from tR to HDFS and RevoHBase provides connectivity from R to HBase. Additionally, RevoHStream allows MapReduce jobs to be developed in R and executed as Hadoop Streaming jobs[6].

6.2 HDFS Overview

To meet these challenges we have to start with some basics. First, we need to understand data storage in Hadoop, how it can be leveraged from R, and why it is important. The basic storage mechanism in Hadoop is HDFS (Hadoop Distributed File System). For an R programmer, being able to read/write files in HDFS from a standalone R Session is the first step in working within the Hadoop ecosystem. Although still bound by the memory constraints of R, this capability allows the analyst to easily work with a data subset and begin some ad hoc analysis without involving outside parties. It also enables the R programmer to store models or other R objects that can then later be recalled and used in MapReduce jobs. When MapReduce jobs finish executing, they normally write their results to HDFS. Inspection of those results and usage for further analysis in R make this functionality essential.

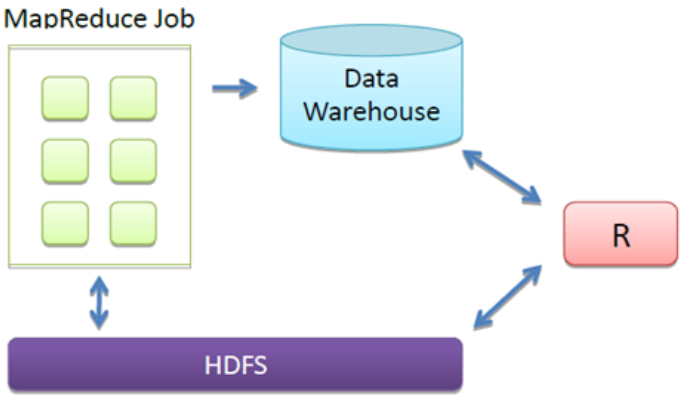
6.3 HBASE Overview

There are several layers that sit on top of HDFS that also provide additional capabilities and make working with HDFS easier. One such implementation is HBASE, Hadoop's answer to providing database like table structures. Just like being able to work with HDFS from inside R, access to HBASE helps open up the Hadoop framework to the R programmer. Although R may not be able to load a billion-row- by-million-column table, working with smaller subsets to perform ad hoc analysis can help lead to solutions that work with the entire data set[7].



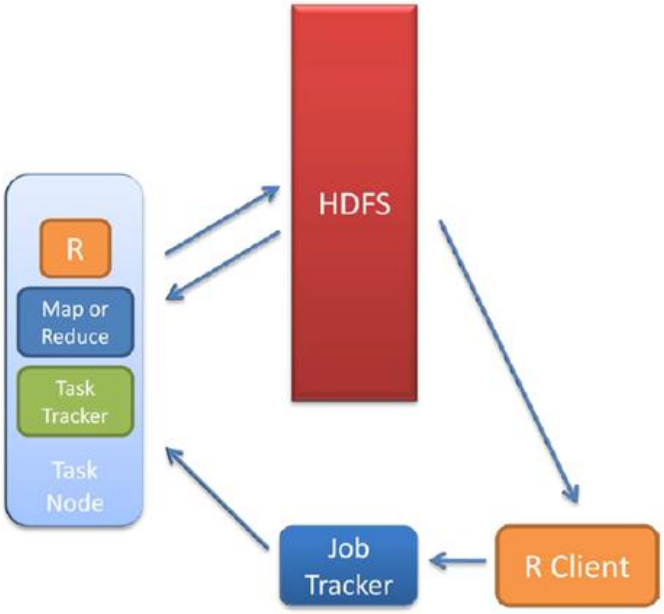
6.4 Map Reduce – Data Reduction

The processing pillar in the Hadoop ecosystem is the MapReduce framework. The framework allows the specification of an operation to be applied to a huge data set, divide the problem and data, and run it in parallel. From an analyst's point of view, this can occur on multiple dimensions. For example, a very large dataset can be reduced into a smaller subset where analytics can be applied. In a traditional data warehousing scenario, this might entail applying an ETL operation on the data to produce something usable by the analyst. In Hadoop, these kinds of operations are written as MapReduce jobs in Java. There are a number of higher level languages like Hive and Pig that make writing these programs easier. The outputs of these jobs can be written back to either HDFS/HBASE or placed in a traditional data warehouse. R can then be used to do the analysis on the data.



6.5 MapReduce - R

Executing R code in the context of a MapReduce job elevates the kinds and size of analytics that can be applied to huge datasets. Problems that fit nicely into this model include "pleasingly parallel" scenarios. Here's a simple use case: Scoring a dataset against a model built in R. This involves pushing the model to the Task nodes in the Hadoop cluster, running a MapReduce job that loads the model into R on a task node, scoring data either row-by row (or in aggregates), and writing the results back to HDFS. In the most simplistic case this can be done with just a Map task. This simulates the "apply" family of operators in R. Other tasks such as quantiles, crosstabs, summaries, data transformations and stochastic calculations (like Monte Carlo simulations) fit well within this paradigm. These implementations don't make any assumptions about how the data is grouped or ordered.Visualizations of huge datasets can provide important insights that help understand the data. Creating a binning algorithm in R that is executed as a MapReduce job can produce an output that can be fed back into an R client to render such visualizations. Other more statistically challenging algorithms can also be implemented in this framework with more effort. These would include data Mining algorithms like K-Means clustering, logistic regression with small numbers of parameters and iterations, and even linear regression.



6.6 Map Reduce – Hybrid

For some kinds of analysis, we can employ a hybrid model that combines using something like HIVE QL, and R. HIVE QL allows us to perform some SQL like capabilities to create naturally occurring groups where R models can be created. As an example, suppose we have some stock ticker data stored in HDFS. If we can use HIVE to partition this data into naturally occurring groups (i.e., stock ticker symbol) we could use R to create a time series model and forecast for each ticker, and do it in parallel. Another possibility might be creating a correlation matrix by using Hive and R, and feeding that into PCA or Factor Analysis routines.

Revolution has created an R package that allows creation of MapReduce jobs in R. The goal is providing a simple and usable interface that allows specification of both Map and Reduce as functions in R. This keeps the data scientist working in R, since he or she does not have to worry about the underlying Hadoop infrastructure. While it's true that the R programmer might have to rethink the approach to how algorithms can be realized and implemented, the potential benefits justify the additional effort.

6.7 Optimizing Algorithms

Finally, there is the approach of developing algorithms that have been explicitly parallelized to run within Hadoop. For example if you wanted to do a linear or logistic regression in R on a 1TB of data stored in HDFS, this requires that the algorithms themselves be implemented in way to use a distributed computing model. Revolution Analytics has a framework for developing these kinds of algorithms to be optimized within Hadoop[8].

VII. Conclusion

The goal of Big Data analytics for security is to obtain actionable intelligence in real time. Although Big Data analytics have significant promise, there are a number of challenges that must be overcome to realize its true potential. The following are only some of the questions that need to be addressed [9]:

1. Data provenance: authenticity and integrity of data used for analytics. As Big Data expands the sources of data it can use, the trustworthiness of each data source needs to be verified and the inclusion of ideas such as adversarial machine learning must be explored in order to identify maliciously inserted data.
2. Privacy: We need regulatory incentives and technical mechanisms to minimize the amount of inferences that Big Data users can make. CSA has a group dedicated to privacy in Big Data and has liaisons with NIST's Big Data working group on security and privacy. We plan to produce new guidelines and white papers exploring the technical means and the best principles for minimizing privacy invasions arising from Big Data analytics.
3. Human-computer interaction: Big Data might facilitate the analysis of diverse sources of data, but a human analyst still has to interpret any result. Compared to the technical mechanisms developed for efficient computation and storage, the human-computer interaction with Big Data has received less attention and this is an area that needs to grow. A good first step in this direction is the use of visualization tools to help analysts understand the data of their systems.

References

[1]. Bryant, R., R. Katz & E. Lazowska. (2008). Big-Data Computing: Creating revolutionary breakthroughs in commerce, science and society. Washington, DC: Computing Community Consortium.

[2]. Camp, J. (2009). Data for Cybersecurity Research: Process and "whish list". Retrieved July 15, 2013, from http://www.gtisc.gatech.edu/files_nsf10/data-wishlist.pdf.

[3]. Cugoala, G. & Margara, A. (2012). Processing Flows of Information: From Data Stream to Complex EventProcessing. ACM Computing Surveys 44, no. 3:15.

[4]. D. Feldman, M. Schmidt, and C. Sohler. Turning big data into tiny data: Constant-size coresets for k-means,pca and projective clustering. In SODA, 2013.

[5]. Apache Hadoop, http://hadoop.apache.org.

[6]. Apache Mahout, http://mahout.apache.org.

[7]. J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu, and G. Fox. Twister: A runtime for iterative MapReduce. In Proc. HPDC, pages 810–818. ACM, 2010.

[8]. Wei Fan and Albert Bifet " Mining Big Data:Current Status and Forecast to the Future",Vol 14,Issue 2,2013

[9]. Xindong Wu , Gong-Quing Wu and Wei Ding " Data Mining with Big data ", IEEE Transactions on Knoweledge and Data Enginnering Vol 26 No1 Jan 2014