Study of Distributed Framework Hadoop and Overview of Machine Learning using Apache Mahout

Raxitkumar Solanki
Department of Computer Science
California State University, Fullerton
Fullerton, CA, USA
raxit65535@csu.fullerton.edu

Sree Harsha Ravilla
Department of Computer Science
California State University, Fullerton
Fullerton, CA, USA
sreeharsha@csu.fullerton.edu

Doina Bein
Department of Computer Science
California State University, Fullerton
Fullerton, CA, USA
dbein@fullerton.edu

***Abstract*—The amount of data generated every day in digital format is overwhelming, so we need storage mechanisms to store it and manage it. The technical solutions to store and manage the data should be scalable to allow extraction of relevant information and analysis. We describe the initial steps on using Apache Mahout to find out the total number of books written by authors of different age groups, analyzing the patterns in the authors' age, and predicting which age group have authored the highest number of books in different calendar years. Publishing houses and literary agents can use our proposed software.**

Keywords- Hadoop; Hadoop Commons; Map Reduce; HDFS (Hadoop distributed file system); Yarn (yet another resource negotiator); Apache Mahout.

I. INTRODUCTION

Traditionally publishing houses and literary agents had to trust their instinct to predict which books would be successful. Many talented writers may never had a chance to see their books on the New York Times bestsellers list due to poor decision-making from the publishers. With the wide usage of big data analytics by many large companies, publishing industry should also consider making use of it. However, what is big data? The first personal computers (PCs) had little memory (e.g., in Kilobytes), still it was manageable for all types of processing tools. However, these days, extremely large amount of data is generated digitally rapidly so we seek tools to manage it [1]. Big data helps in making business decision by uncovering hidden patterns, customer behavior, market trends and other useful information. For building an optimal solution, one should categorize big data as per its five characteristics:

- *Volume*: In big data terminology, the word "big" represent this segment of data.
- *Velocity*: It refers to the rate at which data produces in the real-world environment.
- *Variety*: Big data can be of any type, structured, unstructured, semi-structured, multimedia data, emails, tweets, logs, sensor data, etc.
- *Veracity*: It relates to reliability and genuineness of data source.
- *Value*: This characteristic is important because it gives a reference to the solutions builder, to build solution as per the values of data.

The next step is to define the architecture components semantics and operational model, to build the technical solution. Hadoop is one such framework, well known for its distributed processing and scalability. MapReduce is the commonly implemented Hadoop framework used to process big data and is a programming paradigm used for analyzing massive unstructured data across hundreds and thousands of clustered servers involving two tasks, mapping and reducing:

- In the mapping job, a large unstructured data is taken and converted into partially structured data, where individual elements are broken into key and value pairs.
- In the reducing job the output data from the mapping is given as an input and combines the data tuples into a smaller set of tuples. As the name suggests the reduce job is performed after the mapping job is done.

This paper provides a brief overview of Hadoop and some discussion of data processing and machine learning in a distributed environment data mining.

Books are authored by people of various ages. In this paper, we propose to analyze available datasets of books and authors using MapReduce. MapReduce can help publishers in understanding the pattern in which books have been published in previous years, which can help them in making the right decisions in choosing the authors to be approached for publishing in future years. For our analysis, we will be using the International Standard Book Number (ISBN), which is also known as, author key to identify the unique author ID that helps in gathering information about author and link it up with the book information. Once the data is analyzed, they will help advance the knowledge in business sectors. We use Hadoop and k-means algorithm to analyze and predict the age group of authors who have written a maximum number of books in different years and number of books published every year. We also analyzed the increase in the age group with increase in the years for a period.

The paper is organized as follows. In Section II we discuss Hadoop, followed by Apache Mahout in Section III. The description of the proposed software is given in Section IV. Concluding remarks and future work are given in Section V.

978-1-7281-0554-3/19/$31.00©2019 IEEE

II. Overview of Hadoop

MapReduce job performs parallel processing where the task is distributed among multiple nodes in distributed system. MapReduce came into existence in 2004 and from then it has been used for various analytical purposes in different fields where large amount of data is generated. It allows massive scalability across hundreds or thousands of servers in a Hadoop cluster. The MapReduce concept is simple to understand. It performs two main tasks mapping and reducing. MapReduce is based on Divide and Conquer paradigm that helps us to process the data using different machines. As the data is processed by multiple machine instead of a single machine in parallel, the time taken to process the data is reduced by a tremendous amount. Instead of moving data to the processing unit, we are moving processing unit to the data in the MapReduce Framework that will reduce the strain on a single processing unit.

The concept of a Relational Database Management system (RDBMS) is in existence since the 1970s. The fundamental idea in a relational database is that, it stores data in structured format, and hence it will lead to an easier data retrieval, without burdens of the efficiency of an underlying algorithm. Nevertheless, when it comes to big data, RDBMS does not provide successful management of data, specifically in terms of volume, velocity, and variety.

According to a survey, almost 571 websites are deployed on the web every minute [2], and any search engine, such as Google, likely indexes them. For example, there are more than 307 million people on Facebook. The users post more than 300 million photos every day, 510,000 comments post and 293,000 status update every minute [3]. Google has published white papers on how they are processing this big data in the distributed environment. They use Google File System (GFS), a proprietary technology owned by Google, and MapReduce to index the data on the web. Google had announced MapReduce open source in 2014 [4]. Since GFS is not open source, it became essential to develop the similar technical solution for management of big data. One such technical solution is Hadoop.

Hadoop is a ten years old technology, well known in big data analytics. Doug Cutting and Mike Cafarella developed it. History of Hadoop begins in 1997, at Yahoo; in 2001, it became Apache Lucene, then Apache Nutch, used to rank the web pages. During this development, they have noticed flaws in existing file system [4]. Google published a white paper on GFS in 2003. Cutting and Cafarella took notice of it and developed their own file system, called Nutch Distributed File System (NDFS). NDFS has an underlying idea of distributed processing. The data file divides in multiple data-blocks of 64MB in NDFS, and replication factor for the block was set to three by default. However, the problem of durability of information and recovery and management of failures were still not solved. In 2004, Google made their MapReduce technology open source. NDFS used MapReduce for solving problems like, parallelization, distribution, and fault tolerance. In 2006, Doug Cutting took out MapReduce out of Nutch code base and named it Hadoop. Until 2008, Hadoop was still the subproject of Lucene, hence Cutting did a separate project of Hadoop and licensed under Apache software foundation.

As stated at [5], Hadoop Software library is a framework, which is focused on distributed processing of large datasets over the clusters of computers, using a simple programming model. The clusters are created from commodity hardware. Hadoop scales from a single server to clusters of thousands of machines. Each machine in the cluster offers local storage and computation. Hadoop's library is designed in such a way that, it does not rely on hardware for high availability. It automatically detects the failures and handles it by using the functionality of replication.

Hadoop has four building blocks: Common, Distributed File System, MapReduce, and YARN.

Hadoop Common contains common utilities that support other components, including loading or booting the Hadoop framework from the distribution files.

Hadoop Distributed File System is a Java based file system, which provides high throughput access to the data in it. It provides highly available data storage. In a single cluster, HDFS employs two Name Nodes and multiple Data Nodes. Name Node is visualized as a master server, which has the responsibility of maintaining the namespace tree of data blocks. Data nodes are storage locations on HDFS. A Name Node manages data nodes using heartbeat paradigm. Whenever a file is on HDFS, the underlying algorithm first divides the file into smaller blocks of 64MB, and store the blocks on data nodes in the clusters [5]. Blocks are accessible through its block ID and block pool ID, but the physical location of the block cannot be determined. Further, Hadoop maintains the replica of these smaller blocks. By default, replication in Hadoop is set to three, which means that each data block has three replica blocks in the same cluster. Communication between Name node and data nodes happen through RPC (remote procedure call), and client application communicates with Name node using TCP/IP protocol.

Map Reduce is can perform distributed data processing with actual parallelism, in a Hadoop cluster [6]. It provides fault tolerance in a Hadoop cluster. MapReduce has two components in a single cluster: Job Tracker and Task Tracker. Whenever MapReduce algorithm executed in the cluster, Job Tracker will split the work of mappers and reducers. Task Tracker is responsible for running actual mapper and reducer logic on every machine of the cluster. When nodes in the cluster exceed more than 4000, Map Reduce framework does not behave as expected. Multi-tenancy is one of the tricky issues of MapReduce.

YARN introduced to provide a solution to these problems. YARN act as a cluster resource manager. It gives higher scalability and reliability to Hadoop cluster. YARN divides the Map Reduce task into following components: application master, resource manager, node managers, and containers [7]. The workflow of YARN is as follows:

- Resource manager will be receiving the request from the client to access the application.
- The application will run on the container allocated by the resource manager.
- Resource manager performs negotiations of resources for application through node manager.
- Container will be launched by Node Manager
- Application master executes in the container.

III. Overview of Apache Mahout

Large companies like Facebook, Twitter, LinkedIn, etc. started using Hadoop. Doug Cutting in 2008 decided to separate Hadoop project and license it under Apache. Different companies have different requirements according to their workflow. Hence, they started experimenting on Hadoop, by building their customized solutions on top of Hadoop. All the general-purpose solutions became part of Hadoop and called the Hadoop ecosystem [8].

The success of companies depends on how quickly they can come up with the meaningful information from the huge amount of data. This ideology has engendered the machine-learning paradigm. Netflix uses machine learning to provide video recommendations based on the previous watches. Amazon uses machine learning to provide products recommendation based on previous purchases. Other well know uses of machine learning are in fraud detection for credit cards and email spam detection.

If we want to apply machine learning to the given batch of data on HDFS, then we can use Apache Mahout. It will provide a flexible environment to develop machine-learning algorithm on top of Hadoop framework. Data categorizes into two types, Batch data, and stream data. In simple words, we can visualize batch data as a snapshot of the stream of data. Further, if we see data processing paradigm from an elevated level, data processing can be divided into two categories of distributed and non-distributed data processing (Fig. 1).

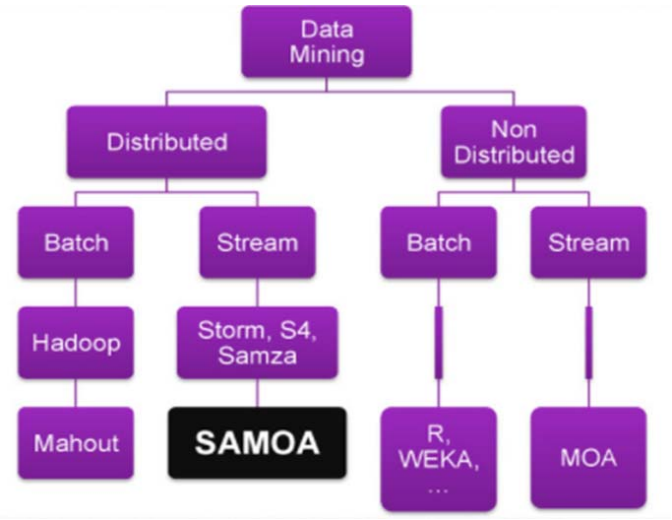

Figure 1. Classification of Data Mining [9]

The purpose of Apache Mahout is to provide a technical solution, which enables the developers to create machine-learning algorithms quickly [10]. Mahout is a simple and reliable framework for building scalable machine learning algorithms. It has pre-implemented algorithms on Scala, Spark, H2o, and Apache Flink, and supports R like syntax for Samsara.

Apache Mahout supports the following categories of machine learning algorithms:

- Canopy: a simple and fast clustering algorithm, in which all the objects represented in multidimensional space. The idea is based on fast approximate distance. Canopy clustering setup initial seeds in clustering algorithms like k-means. Setting up initial seeds or initial clustering, will reduce expensive distance measurements, by ignoring the points outside canopy.
- K-means: is a very popular clustering algorithm that clusters items into k clusters based on the distance of the items from the centroid of the data set.
- Fuzzy K-means: an extension of k-means algorithm that allows an item to belong to more than one cluster.
- Logistic Regression: a classification algorithm, used to find probability of occurrence of an event. It is the standard algorithm in industry; build potentially in fraud detecting and advertising quality products.
- Naive Bayes: a classification algorithm, which uses probabilistic model for classification. It has been a standard in the industry, for classification of text. Mahout supports two Naive Bayes builds. Multinomial naive bays (Bayes), and Transformed weight-normalized complement. (CBayes). CBayes is extension of Bayes, which performs well on data sets with skewed classes.
- Collaborative filtering: makes automatic predictions about the user interests by analyzing the historical data. Mahout supports following collaborative filtering algorithms [10]: user-based collaborative filtering, item based collaborative filtering, matrix factorization with ALS, matrix factorization with ALS on implicit feedback, weighted matrix factorization and SVD++.
- Frequent pattern mining: group the items together, based on frequency of occurrences of any item.

Data mining is used by most of the online shopping applications, to market other products to users, based on users' previous purchase history [11]. Fraud detection in the banking application and loan strategies of users can be estimated using data mining. Crime detection and criminal suspect identification improved by using data mining technologies in law enforcement. Manufacturing process became more meaningful after deciding what to manufacture, by analyzing the previous use of product using predictive data analytics.

However, data mining concept has potential privacy issues. Data collected for data mining, can have two outcomes,

either it can give some meaningful information, which is beneficial to everyone in the world, or it can give significant information again, but it will be beneficial to only few people, who purposely mined the data to hamper privacy of data owners, by selling the results of data mining.

Security is also a considerable issue in data mining. It is essential that any organization, who have sensitive information about their customers/users, they should maintain the decorum and keep that information private and secure.

IV. Proposed Software

In the past literary agents and publishing houses had to trust their intuition in choosing the authors and publish their books, leaving out many authors unattended. Our proposed software based on data mining will help the publishers making decisions that are more informed. For data mining, we acquired several datasets from Open Library Data Dump, and one book reviews dataset from Book-Crossing Dataset.

We can find out the total number of books written by authors of different age groups, analyzing the patterns in the ages of authors, and predicting which age group of authors have written the highest number of books in different calendar years. Five steps are considered for the system architecture (Fig. 2).

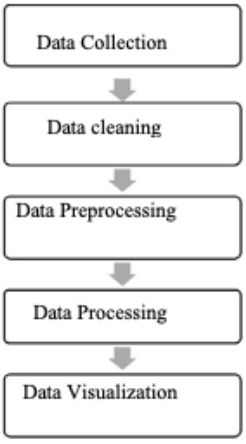

Figure 2. Main tasks of proposed software

The software starts with collecting data and it goes on to the next phase where we clean, process and prepare the data for the next phase of data visualization. The architecture ends up with the phase of developing a model that accurately predicts the performance of the student. By following these steps, we end up having a graphical model that helps us analyze the pattern and predict the future publishing scenario.

First, we will analyze the data that we have, to find correlations between various datasets. Then, we decide on certain analyses that can be performed with correlations that we obtained. We considered these two correlations:

1. Published date of the book, the birthdate of the author
2. Genre of the book, location where the book was published.

Using those three correlations, we propose to perform the following analyses:

1. For various years, how many books have been published by authors of various ages. So we are looking at counts of books across various ages of authors, and how that count of books has evolved over years. We will be representing for each year, authors of what age has written the maximum number of books. For a particular year, we will also look at counts of books written across different age groups. For example, count of books by authors of age below 30, between 30-60, and above 60.
2. Since there are more than 200 unique genres in the book dataset, and we will compute the genre with maximum number of books. Therefore, we will dividing it into clusters and finding out whether that cluster has the maximum number of books.

The dataset of [12] contains information of all the latest editions of books (Over 26GB of Editions related data) that are published over many years ranging from 16th century to 2014. Each record contains meta-data of edition related information like Book ISBN, title, genre, published place, published date, publishers, languages, subject, number of pages, author key, sub-title, series, publish country, contributions etc. It is stored in JSON format and has over 26GB of book related information with varied number of columns for each edition. But there are some common attributes for all the editions that are identified above. We will be using a subset of these attributes like ISBN, published date, published country, author key, and genre for various tasks.

The collected JSON files have been converted to CSV files where we take the required data and nullify the unwanted data. The data cleaning is done by performing mapping job using MapReduce.

From the two CSV files generated by cleaning, we use the author key, which is a unique key given to authors to perform the mapping job. Required information from both the files are mapped together with the help of a unique key. This file generated by mapping is used as an intermediate file with all the required information. The intermediate file generated from the preprocessing step is used as an input for performing the reduce job. The final output consists for years, age group of authors who have written maximum number of books every year and total number of published books every year.

Another interesting attribute in the books' dataset is the genre attribute of each book. It would be interesting idea to determine, how publishers are dealing with genres at different locations in the United States. We have data from other countries as well, but the country codes and the published places are skewed, and that makes the preprocessing much more difficult.

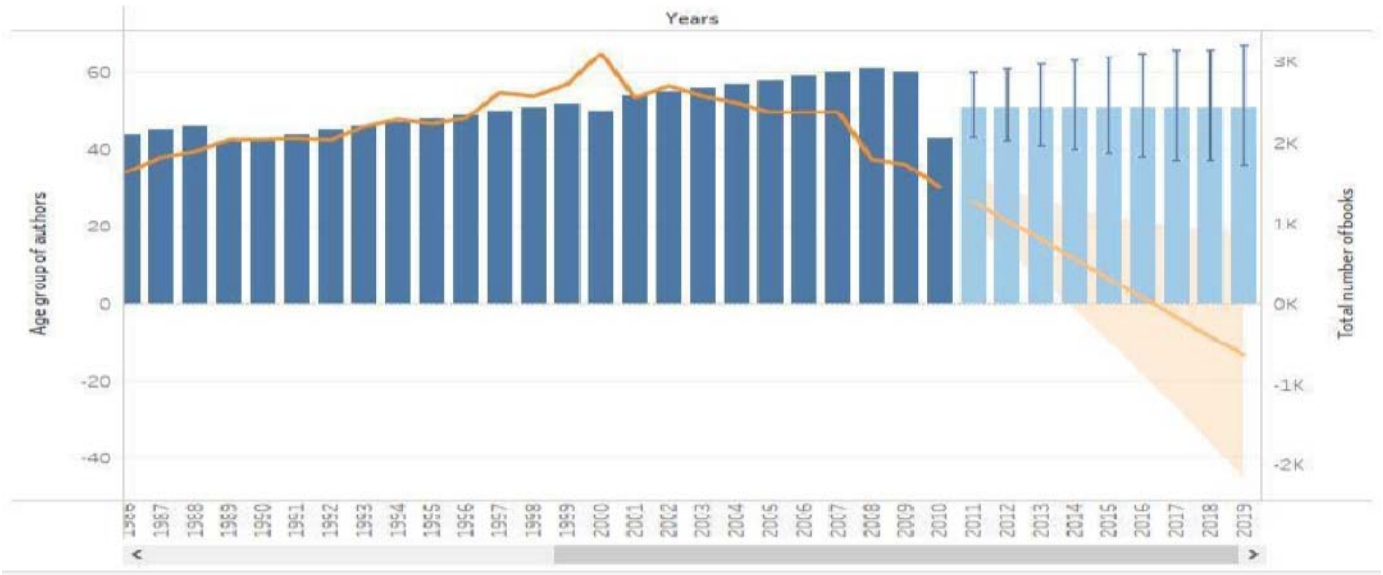

Figure 3. Number of books published and age of authors

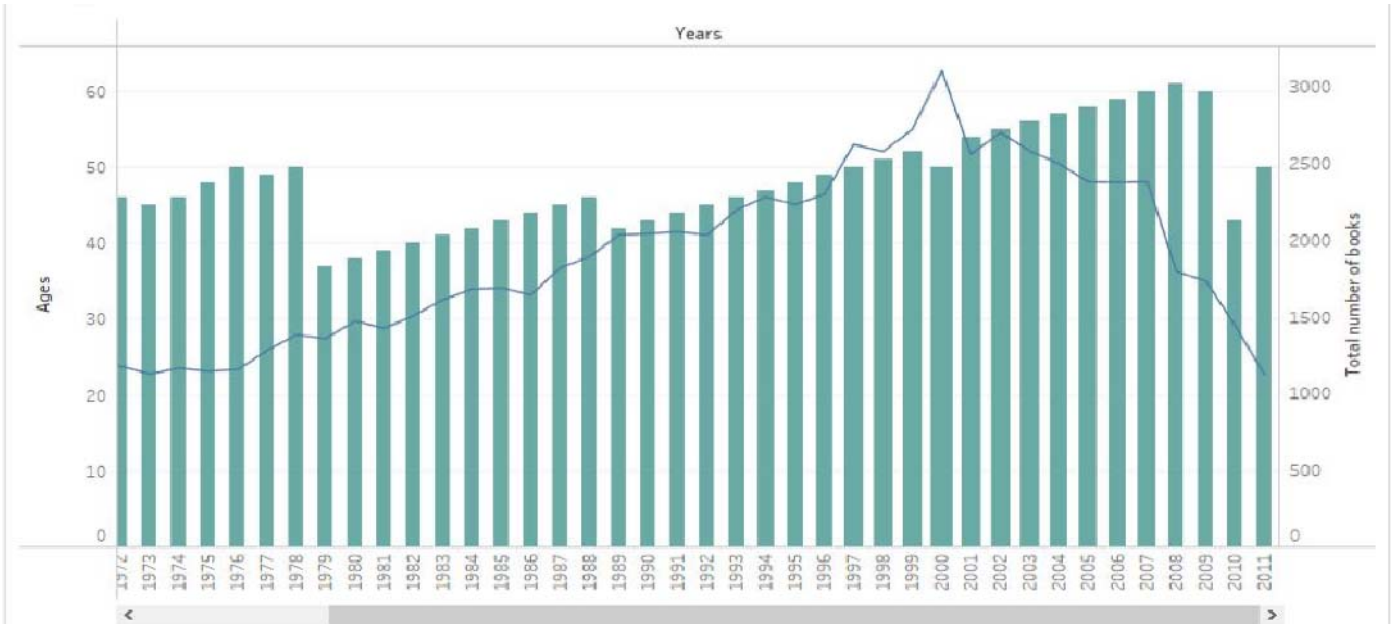

Figure 4. Age group of authors publishing the maximum number of books

We are interested in two values from editions data dump.

1. Genre associated with the book

2. Published place, where the book is published in the US

For analysis, we divided genres into two groups, fiction and non-fiction. We wanted to see how these two groups perform over multiple locations in the US. For cluster, we experimented by dividing the US into two groups (Eastern and Western) and then into four groups (Eastern, Western, Northern and Southern). So the number clusters need to be defined based on the experiment chosen. We choose New York (eastern), Atlanta (southern), California (western) and Indiana (northern) as endpoints for our clusters.

When we chose two clusters, non-fiction seems to be dominant over the following two regions: eastern United States and western United States. When we chose four clusters, all four regions seemed to have non-fiction as a dominating genre.

Based on the data collected from Open Library Data Dump, we have generated a dataset with the age group of authors who have written the highest number of books and the total number of published books every year. The data is represented in the graphical format; we also generated a forecast showing the number of books to be published in future years and in which age group of authors books will be published (Fig. 3).

With help of this graphical pattern generated by applying the MapReduce we can analyze a pattern in which books were publishes in previous years. We can see the raise in the age group of authors with the passing years. We can conclude that publishers might be approaching same authors repeatedly. Therefore, when they stop writing the books, there might be fall in the number of published books in future years. We analyzed the age group of authors who have written a maximum number of books in different years and number of books published every year (Fig. 4). The graph in Fig. 4 represents the results of the data analysis and the prediction of the analysis in which the increase in the age group with increase in the years for a period. Which means that, the same authors might be writing books every year for a particular period and the cycle is repeating for almost every 10 years.

V. Conclusion and Future Work

The paper give details of Hadoop framework and Apache Mahout. We use Hadoop and k-means algorithm to analyze and predict the age group of authors who have written a maximum number of books in different years and number of books published every year. We notice a cycle that repeats for almost every 10 years. From this information, we can conclude that publishers are mostly focusing on the same authors who have published books in past. This also means that new authors must have potentially not be considered for publication. Using this analysis, a publisher can understand the pattern in which books were published in previous years. For a new author, this analysis can help establishing new strategies to get his/her books accepted for publication.

We performed a forecast using Tableau that shows a drop in the number of published books in future years if the pattern is not changed. We also have obtained from the software that the genre with maximum number of books is fiction in United States. Such result was obtained by dividing the datasets into two clusters, one for genre fiction or the other for non-fiction, and we obtained the cluster with the maximum number of books is fiction.

As future analysis, we propose to consider the ratings of the authors by taking the mean and standard deviation of the ratings and finding out: (i) how many authors fall above the mean and below the standard deviation, (ii) how many authors fall above the mean and the standard deviation, (iii) how many authors are below the mean and above the standard deviation, and (iv) how many authors fall below the mean and the standard deviation.

References

[1] X. Qin , “Making Use of the Big Data: Next Generation of Algorithm Trading”, vol. 7530, pp. 34-41, 2012.

[2] H. Mousnif, H. Sabah, O.S. Younes, and Y. Douiji, “From Big Data to Big Projects: A Step-by-Step Roadmap,” Proceedings of 2014 International Conference on Future Internet of Things and Cloud, (FiCloud 2014).

[3] Forbes, “How Much Data Do We Create Every Day?”, available online at https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#787667c960ba, last accessed December 20, 2018.

[4] A.K Jogawath, “History of Hadoop and Map Reduce”, 2015, available online at https://ajaykumarjogawath.wordpress.com/2015/09/22/history-of-hadoop-and-map-reduce/, last accessed December 20, 2018.

[5] Apache Hadoop, “What is Apache Hadoop?”, available online at http://hadoop.apache.org/, last accessed December 20, 2018.

[6] Apache Hadoop, “MapReduce Tutorial”, available online at https://hadoop.apache.org/docs/r2.7.0/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html, last accessed December 20, 2018.

[7] T. Jones and M. Nelson, IBM Developer, Moving ahead with Hadoop YARN”, 2013, available online at https://www.ibm.com/developerworks /library/bd-hadoopyarn/, last accessed December 20, 2018.

[8] Jay Kaycee, “The Big Data Block-Hadoop Ecosystem Overview,” available online at http://thebigdatablog.weebly.com/blog/the-hadoop-ecosystem-overview, last accessed December 20, 2018.

[9] G. De Francisci Morales, A. Bifet. "SAMOA: Scalable Advanced Massive Online Analysis." Journal of Machine Learning Research, vol.16 (Jan), pp.149-153, 2015.

[9] Apache Mahout, “What is Apache Mahout?,” available online at http://mahout.apache.org/, last accessed December 20, 2018.

[10] Zentut, “Advantages and Disadvantages of Data Mining,”, 2018, available online at http://www.zentut.com/data-mining/advantages-and-disadvantages-of-data-mining/, last accessed December 20, 2018.

[11] Open Library, file located at http://openlibrary.org/data/ol_dump_editions_latest.txt.gz, last accessed December 20, 2018.