FAST DATA CLUSTERING AND OUTLIER DETECTION USING K-MEANS CLUSTERING ON APACHE SPARK

[1]YADIGAR ERDEM, [2]CANER OZCAN

[1,2]Department of Computer Engineering, Karabuk University, Karabuk, Turkey
E-mail: [1]ydgrerdem@gmail.com, [2]canerozcan@karabuk.edu.tr

Abstract— The components forming the information society nowadays are seen in all areas of our lives. As computers have a great deal of importance in our lives, the amount of information has begun to gather meaningful and specific qualities. Not only the amount of information is increased, but also the speed of access to information has increased. Large data is the transformed form of all data recovered from different sources such as social media sharing, network blogs, photos, videos, log files, etc. into a meaningful and workable forms. Clustering on Big Data with machine learning methods is very useful. Clustering process allows very similar data to be placed under a group by separating the data into a specific group. Once datasets are divided, outlier detection is used to find fraudulent data. In this study, it is aimed to make data clustering and outlier detection process faster by using Apache Spark technology on Big Data with K-means clustering method. Clustering on Big Data can be time consuming. For this reason, Apache Spark fast cluster computing architecture is used in this study. It is aimed to perform fault tolerant, reliable, consistent and fast clustering process using this technology. The MLlib library of Spark components has a relatively small code size and ease of use. Its goal is to make practical machine learning scalable and useful. K-means method, which is included in the MLlib library used in this study, provides a successful analysis of big data. The results are presented in tables and graphs using sample dataset.

Index Terms— Apache Spark, Big Data, K-means Clustering, Outlier Detection.

I. INTRODUCTION

With the advancement of technology and the development of the internet, the power of knowledge has also come to the forefront today, and many of the phenomena in the internet world have begun to be referred to as information dump. Software companies thought that meaningful data could come from this dump, and by doing so, they found out what we call Big Data. The definition does not just mean "data that takes up too much space on the disk". It also means that data cannot be processed with traditional methods and tools. Big data is a form that has been transformed into meaningful and processable data obtained from different sources such as social media shares, photo archives, continuously recorded 'log' files [1]. Big Data processing is the result of rapid improvements in volume and velocity of data produced by those applications.

Content, domain and samples of the big data is reviewed in detail and its advantages and challenges have been examined [1]. Big data applications need to be analyzed and appropriately executed to achieve more successful results. Along with the developing technology, Internet of Things (IoT) applications have found a wide development environment and various applications have been realized. The current research of IoT and the major IoT applications in industries have been examined and future trends and challenges are reviewed [2]. A theorem that qualifies the features of the Big Data revolution from the data mining perspective is presented and the challenging topics in the data-driven model are analyzed [3].

Big data analytics is the area of research to discover patterns, relations and extract information from data. This is achieved by using supervised and unsupervised machine learning algorithms, often learned from existing data. Clustering is one of the solutions of unsupervised learning problems and the task of grouping a set of objects in such a way that similar objects are grouped under the same group [4]. Clustering is useful in machine learning situations and particularly appropriate for the exploration of relationships among the data points to make a preliminary evaluation of their structure [5]. K-means is one of the simplest and fast unsupervised learning algorithms that produces solutions to the well-known clustering problem. This technique is efficient for processing large datasets. A simple and effective application of the K-means clustering algorithm is presented in [6]. An overview of clustering, summarize general clustering methods, discuss the key challenges and major issues in designing clustering algorithms, and point out some of the useful research directions are provided in [7]. Modified versions of the K-means algorithm [8]-[10] are presented in recent works. A novel algorithm [8] which adopts a new nonmetric distance measure based on point symmetry idea is proposed. A clustering algorithm based on a standard K-means approach that does not require user parameter specification is also presented in [9]. Distance measure and k-means-based algorithm for circular invariant clustering of vectors containing directional information is introduced [10].

Different techniques for outlier detection have been proposed over the years. The prominence of anomaly detection comes from the fact that abnormalities in the data are transformed into valuable information in many different application areas. K-means method is generalized for clustering data and discovering outliers [11]. As a result of this study, the method was found to be very sensitive to outliers. K-means clustering is also used for credit card fraud detection [12], financial fraud detection [13], medical diagnosis [14] and refund fraud detection [15].

However, these algorithms are computationally expensive, either in the clustering or outlier detection phases. Alternatively, parallel design approaches have been developed in order to increase processing speeds. This implies that technologies that explicitly support distributed computations must be used. Spark-based intelligent k-means methods [16], [17] are designed for big data clustering. In a similar study [18], K-means-based clustering implementation which includes two iterative processes: distance calculation and centroid updating is parallelized on Spark. K-Means clustering algorithm in which automated for the number of clusters [19] is presented using big data. In another study [20], performance based on in-memory and on-disk computation models of distributed K-Means clustering is analyzed.

The remainder of this paper presented as follows: Section II explains problem domain. The overall framework for parallelizing K-Means algorithm is explained in Section III. Section IV explains SPARK open-source cluster computing framework which is used to process our data. Section V explains the experimental results and analysis of the research and its discussion. Finally, the paper is concluded in Section VI.

II. PROBLEM DEFINITION

The traditional algorithms and tools used for data management are insufficient to process big data. They are not able to provide effective solutions in managing the growing data every day. Obtaining knowledge from these data environments and managing the acquired knowledge are among the most popular in information processing research. More powerful and intelligent machine learning methods which have the ability to extract knowledge from the data are developed. In this study, individual household electric power consumption dataset [21] is chosen and analyzed.

Clustering methods are used to find new and substantial information in a particular set of data. Within the clustering methods, K-means can be denoted as one of the earliest methods. This method is well-known and efficient for processing large datasets. Using clustering, datasets are partitioned and outlier detection is used to find the fraud data. An outlier is a record of an expected behavior which does not fit well with a generic pattern in a given dataset. Outlier detection in one-dimensional data is given in Fig. 1.

The development of high-level clustering programming models has led to the emergence of parallel implementations of K-means methods. Spark is preferred because it is very convenient for parallelization of iterative algorithms such as K-means. A variety of machine learning algorithms have been developed in Spark, including the K-means under the MLlib library. Therefore an effective clustering framework is used to solve our problem.

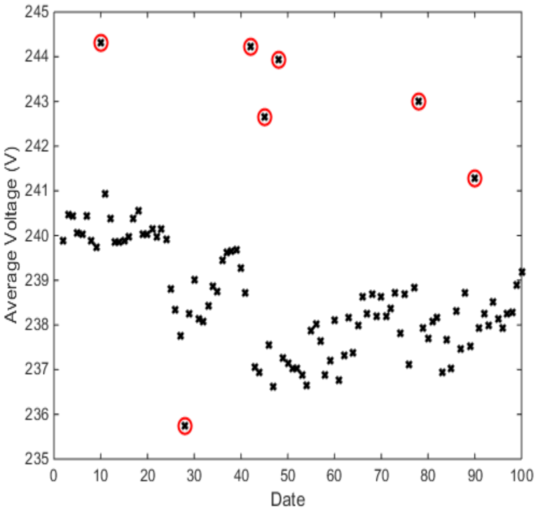


Fig. 1. Outlier Detection in One-Dimensional Data.

III. SYSTEM ARCHITECTURE AND METHODS USED

We have performed our work in two stages:

Stage I- Clustering process using K-means

Stage II- Outlier or anomaly detection process using distance based approach

Clustering is an important tool for outlier analysis. Using the clustering technique as a first step, the data is divided into groups with similar characteristics. Centroids are calculated for each dataset group. The maximum distance value is computed for each cluster using the distance based method. Then, these values examine and compare with threshold value which is taken by user. If the maximum distance value is greater than the threshold, data is considered as outlier. But if the opposite condition is true, data is considered as inlier namely real object. Distance based methods are much more efficient and useful. These methods may not require a detailed understanding of their application areas. Additionally, if they have a distance measure, they can be defined for any data type. Outlier detection is an important data analysis task and it can be applied to various application areas. Outliers in the data point to something important that cannot be ignored and ensure useful patterns for further analysis. Outlier detection is the identification of items or objects which are considerably inconsistent with other items in a dataset or which are far away from their cluster centers. System architecture is shown in Fig. 2. One of the simplest tasks we can perform on an unlabeled dataset is to find groups of data that similar each other in the dataset. K-means [22] is one of the most popular and simplest unsupervised learning algorithms that solve the well-known clustering problem. Two important assumptions are that each input can belong to only one cluster, and user already knows how many clusters exist.

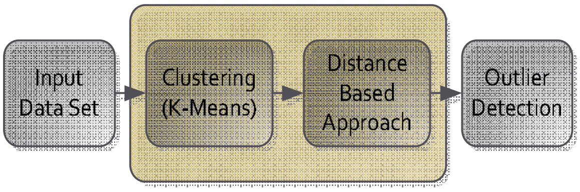


Fig. 2. System Architecture.

K-Means method is numerical, unsupervised and process iteratively. Unsupervised technique is useful when there is no prior knowledge about the particular class of observations in a dataset. In the K-Means clustering method, n objects are divided into k clusters, allowing each object belongs to the nearest mean of the clusters. At this point k different clusters will be created with the greatest possible variation. Since the number of clusters providing the greatest distance is not known in advance, it has to be calculated from the data. Finally, the aim of this algorithm is to minimize an objective function, in this case a squared error function. The objective function:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \qquad (1)$$

where k is the number of clusters and n is the number of cases. $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the centroid of cluster c_j. The algorithm is composed of the following steps. The first step of the K-Means procedure is to place the K points in the area represented by the clustered objects. These points act as the initial cluster centroids. The second step is to assign each data point to the group that has the closest centroid. In the third step the positions of the K centroids are calculated, and a new partition is created. Step 2 and 3 are then repeated until there is no change in the centers. It works in an iterative manner and the cluster centers get updated at the end of each iteration. The main objective of the K-Means algorithm is to minimize the sum of the squared error for all the clusters [7]. Pseudo code of K-Means algorithm is sketched in Alg. 1.

Algorithm 1 K-Means

```
  Set $\vec{c}_i, \dots, \vec{c}_k$ to be distinct randomly selected inputs from
  $\vec{x}_1, \dots, \vec{x}_n$
  repeat
    for $i = 1$ to $n$
      $\gamma_{ij} = \begin{cases} 1 & \text{if } j = \mathrm{argmin}_j \|\vec{x}_j - \vec{c}_j\|^2 \\ 0 & \text{otherwise} \end{cases}$
    end for
    for $j = 1$ to $k$
      $n_j = \sum_{i=1}^{n} \gamma_{ij}$
      $c_j = \frac{1}{n_j} \sum_{i=1}^{n} \gamma_{ij} \vec{x}_i$
    end for
  until convergence
  Return $\vec{c}_1, \dots, \vec{c}_k$
```

Since the number of clusters at the beginning is not known, it can be estimated empirically. To solve this situation, the results of multiple runs for different cluster numbers can be compared and then the best one is selected based on a predefined criterion. It will be seen that a large cluster will probably reduce the error, but increase the risk of overfitting. The K-Means algorithm may not always find the most appropriate result corresponding to the objective function. However, the algorithm is very sensitive to randomly selected initial centers. To reduce this effect, the K-Means algorithm can be run multiple times [23].

IV. PROJECT DESIGN

Even though Big Data is a very new technology, it offers a wide range of tools to compute big data. The most important and widely used of these applications are the Hadoop software tools. Hadoop is a scalable, fault-tolerant and open source library developed by Java that runs applications for handling large amounts of data on a cluster of commodity hardware [24]. It combines Hadoop MapReduce features with a distributed file system called Hadoop Distributed File System (HDFS). Through HDFS, disks from ordinary servers come together to form a large, single virtual disk. In this way, many files of very large size can be stored in this file system. These files are distributed in blocks (default 64MB) on multiple and different servers.

On the other hand, Hadoop MapReduce is a way to handle large files on HDFS. The program, which consists of the Map function used to filter the data you want and the Reduce functions that allow you to get results from these data, is run on Hadoop. Hadoop is responsible for distributing the Map and Reduce threads over the cluster and processing them at the same time, and reassembling the resulting data [17]. It can be said that the power of Hadoop comes from the fact that the processed files are always linearly scaled by reading the corresponding node from the local disk, not engaging in network traffic, and handling multiple jobs at the same time. Basically Hadoop distributes files via HDFS and then processes them through the MapReduce.

Apache Spark is a new platform which works on Hadoop. Apache Spark is an open source library developed by Scala that enables us to work in parallel on large datasets [17]. Spark is a member of the Hadoop family and make up some of the shortcomings of Hadoop's weaknesses rather than being a substitute for Hadoop. It's easy to use and projects can be developed with less effort compared to MapReduce. Spark processes the data as in-memory efficiently and is designed to get faster results. Spark in-memory data sharing is represented in Fig. 3. Java and Python languages are also supported with Spark. One of the most useful features is that it becomes very easy to develop with the abstraction that Spark brings. It is provided with Resilient Distributed Dataset (RDD) as if it were a collection object. RDDs include action methods such as reduce, collect, count, distinct as well as many transformation methods such as map, filter, union, cartesian, join, groupByKey and sortByKey which will facilitate data processing. A file or a collection object is converted into RDD and the data is processed by calling these methods sequentially.

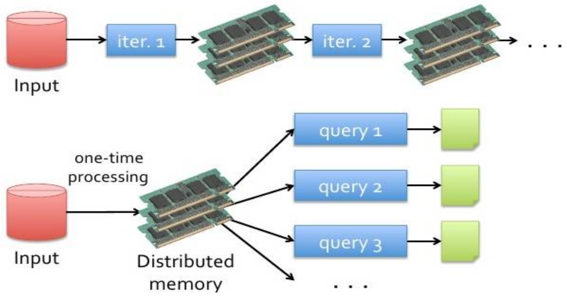


Fig. 3. Spark In-Memory Data Sharing [25].

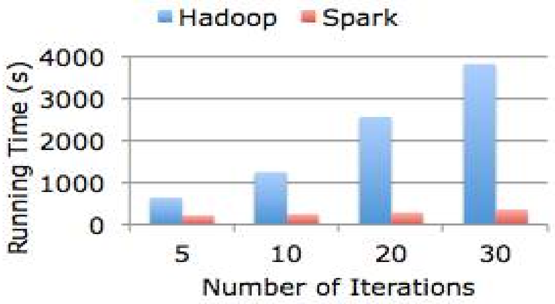


Fig. 4. Logistic Regression in Hadoop and Spark [26].

Spark technology provides a machine learning library for detailed processing and analysis of big data. MLlib includes high-quality algorithms for clustering, classification, regression, dimensionality reduction, training and other statistical applications. MLlib can approximately run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk. Running time according to the number of iterations of logistic regression in Hadoop and Spark is shown in Fig. 3. A parallelized version of the K-Means method named K-Means|| is included in the Spark MLlib implementation. In our study, we performed analyzes using this algorithm. K-Means|| algorithm in MLlib has some parameters given in Table 1.

Table 1. Parameters of K-Means Algorithm in MLlib.

Parameters	Explanations
k	number of desired clusters
maxIterations	maximum number of iterations to run
initializationMode	random initialization via k-means
runs	no effect since Spark 2.0.0
initializationSteps	number of steps in the k-means algorithm
epsilon	distance threshold within which we consider k-means to have converged
initialModel	optional set of cluster centers used for initialization

V. EXPERIMENTAL RESULTS AND ANALYSIS

This section presents experimental results obtained by the studies carried out. In experimental studies, a computer with Intel i7 3.4 GHz 4 core 8 GB RAM is used. The speed factor has been increased with SSD, and 13 GB memory on 16 GB memory is presented on the JVM so that Spark can use enough space for memory analysis. For the clustering operation, which is the priority step, the number of clusters and the number of iterations are determined as 10 and 20, respectively, and the selected K-Means algorithm is executed. After the clusters were identified, distance calculation was performed for outlier detection. The result of the outlier detection is that it is possible to detect a momentary problem in electricity consumption.

Sample data is taken from UCI machine learning repository that provided various types of datasets. This dataset can be used for clustering and regression. Number of attributes is nine and number of instances is 2075259. Dataset has real value attributes and has missing values. In order to eliminate these values, dataset has been preprocessed so that a total of 25979 entries are deleted. At this point, a more complete study has been done.

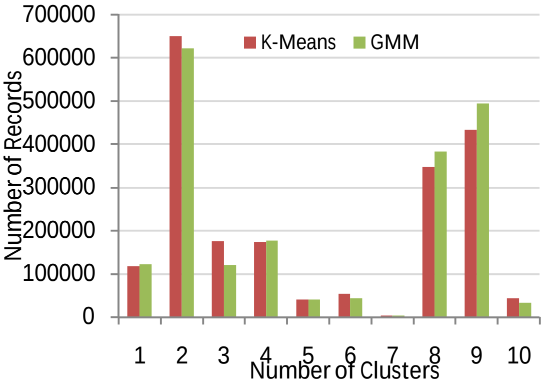
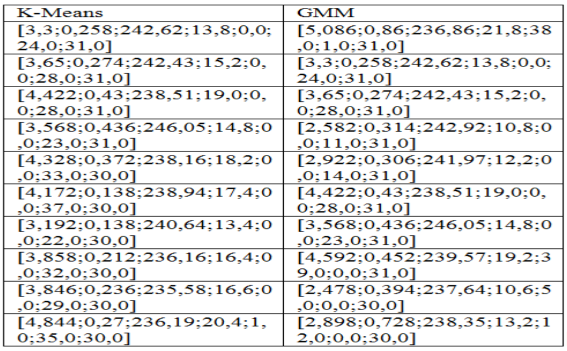


Fig. 5. Clustering results of K-Means and GMM

The Gaussian Mixture Model (GMM), which is also implemented in the MLlib, is also used to reveal the superiority of the K-means method. GMM provides a compound distribution from which each of the points is taken from one of its own probable k Gaussian sub-distributions. Clustering results using K-means and GMM are given in Fig. 5. According to results, it is observed that similar clusters are formed as a result of these two methods. When the cluster information is examined in detail, it is found that different patterns can be recognized for household power consumption.

After applying clustering operation, an analysis was performed to determine the outliers in the data set. Using distance based approach; the desired outlier data can be detected. Table 2 shows the values of 10 outliers data detected using K-means and GMM on sample dataset. The fact that some of these rows are the same indicates that both methods detect the same records as outliers. Running times according to different data sizes for K-Means Spark, K-Means KNIME and GMM Spark are measured as given in Fig. 6. Each 'K' represents 1000 records for a given section. As seen in Fig 6, K-Means algorithm is much faster than both Gauss method on the Spark and K-Means in the KNIME framework due to its efficient numerical schema. As the data grow exponentially, the success of the K-Means method increases more.

Table 2. Found 10 Outliers Data using K-Means and GMM.



K-Means	GMM
[3,3;0,258;242,62;13,8;0,0;24,0;31,0]	[5,086;0,86;236,86;21,8;38,0;1,0;31,0]
[3,65;0,274;242,43;15,2;0,0;28,0;31,0]	[3,3;0,258;242,62;13,8;0,0;24,0;31,0]
[4,422;0,43;238,51;19,0;0,0;28,0;31,0]	[3,65;0,274;242,43;15,2;0,0;28,0;31,0]
[3,568;0,436;246,05;14,8;0,0;23,0;31,0]	[2,582;0,314;242,92;10,8;0,0;11,0;31,0]
[4,328;0,372;238,16;18,2;0,0;33,0;30,0]	[2,922;0,306;241,97;12,2;0,0;14,0;31,0]
[4,172;0,138;238,94;17,4;0,0;37,0;30,0]	[4,422;0,43;238,51;19,0;0,0;28,0;31,0]
[3,192;0,138;240,64;13,4;0,0;22,0;30,0]	[3,568;0,436;246,05;14,8;0,0;23,0;31,0]
[3,858;0,212;236,16;16,4;0,0;32,0;30,0]	[4,592;0,452;239,57;19,2;39,0;0,0;31,0]
[3,846;0,236;235,58;16,6;0,0;29,0;30,0]	[2,478;0,394;237,64;10,6;5,0;0,0;30,0]
[4,844;0,27;236,19;20,4;1,0;35,0;30,0]	[2,898;0,728;238,35;13,2;12,0;0,0;30,0]

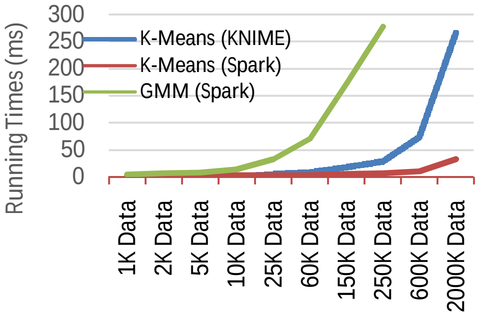


Fig. 6. Running Times Based on Different Data Sizes.

CONCLUSION

In this study, data clustering and outlier detection process is realized faster by using Spark technology on Big Data. Experimental results prove that outlier detection is successfully achieved after using the K-Means algorithm implemented in Spark MLlib. Once datasets are clustered, outlier detection is applied efficiently on individual household electric power consumption datasets to find fraudulent data. Moreover the algorithm scales gracefully on increasing the data size. Experiments on UCI Machine Learning Repository dataset demonstrate the effectiveness of our study. This study can be extended for large scale real time streaming datasets.

REFERENCES

[1] S. Sagiroglu and D. Sinanc, “Big data: A review,” in Collaboration Technologies and Systems (CTS), International Conference on. IEEE, pp. 42-47, 2013.

[2] L. D. Xu, W. He, S. Li, “Internet of things in industries: A survey,” IEEE Trans. Ind. Informat., vol. 10, no. 4, pp. 2233-2243, Nov. 2014.

[3] X.Wu, X.Zhu, G.Q.Wu, W.Ding, “Data mining with big data,” IEEE Trans. Knowl. Data Eng. vol. 26, no. 1, pp. 97-107, 2014.

[4] Rui Xu, D. Wunsch, “Survey of clustering algorithms,” IEEE Transactions on Neural Networks, vol. 16, no. 3, pp. 645-678, 2005.

[5] A.K. Jain, M.N. Murty, and P.J. Flynn, “Data Clustering: A Review,” ACM Computing Surveys, vol. 31, no. 3, pp. 2645-323, Sep. 1999.

[6] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, A. Y. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 881-892, July 2002.

[7] A.K. Jain, “Data clustering: 50 years beyond K-means,” Pattern Recognition Letters, vol. 31, no. 8, pp. 651-666, 2010.

[8] M.-C. Su and C.-H. Chou, “A modified version of the Kmeans algorithm with a distance based on cluster symmetry,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 23, no. 6, pp. 674-680, 2001.

[9] J. G. Wilpon, L. R. Rabiner, “A modified K-means clustering algorithm for use in isolated word recognition,” IEEE Trans. Acoust. Speech Signal Process., vol. 33, no. 3, pp. 587-594, Jun. 1985.

[10] D. Charalampidis, “A modified K-means algorithm for circular invariant clustering,” IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 12, pp. 1856-65, 2005.

[11] S. Chawla, and A. Gionis, “k-means--: A unified approach to clustering and outlier detection,” 13th SIAM International Conference on Data Mining, pp. 189-197, 2013.

[12] M. Singh, Aashima, S. Raheja, “Credit Card Fraud Recognition by Modifying K-means,” IJARCSSE, vol. 4, no. 5, pp. 1291-1296, May 2014.

[13] A. Sorin Sabau, “Survey of clustering based Financial Fraud Detection Research,” Informatica Economica, vol. 16, no. 1, 2012.

[14] Ms. S. D. Pachgade, Ms. S. S. Dhande, “Outlier Detection over Data Set Using Cluster-Based and Distance-Based Approach”, IJARCSSE, vol 2, no. 6, pp. 12-16, Jun. 2012.

[15] H.Issa, M. A. Vasarhelyi, “Application of Anomaly Detection Techniques to Identify Fraudulent Refunds,” Aug. 2011.

[16] I. Kusuma, M. A. Ma'sum, N. Habibie, W. Jatmiko, H. Suhartanto, “Design of intelligent k-means based on spark for big data clustering,” International Workshop on Big Data and Information Security (IWBIS), pp. 89-96, 2016.

[17] X. Mallios, V. Vassalos, T. Venetis, A. Vlachou, “A Framework for Clustering and Classification of Big Data Using Spark,” On the Move to Meaningful Internet Systems: OTM 2016 Conferences, Lecture Notes in Computer Science, vol. 10033, 2016.

[18] B. Wang, J. Yin, Q. Hua, Z. Wu, J. Cao, “Parallelizing K-Means-Based Clustering on Spark,” Advanced Cloud and Big Data (CBD), pp. 31-36, 2016.

[19] A. Sinha, P. K. Jana, “A novel K-means based clustering algorithm for big data,” Advances in Computing, Communications and Informatics (ICACCI), pp. 1875-1879, 2016.

[20] S. Ketu, S. Agarwal, “Performance enhancement of distributed K-Means clustering for big Data analytics through in-memory computation,” Contemporary Computing (IC3), 2015.

[21] M. Lichman, “UCI Machine Learning Repository,” [http://archive.ics.uci.edu/ml]. Irvine, University of California, School of Information and Computer Science, 2013.

[22] J. B. MacQueen, “Some Methods for classification and Analysis of Multivariate Observations,” Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, vol. 1, pp. 281-297, 1967.

[23] M. Dunham, “Data Mining: Introductory and Advanced Topics,” Prentice Hall, USA, 2003.

[24] J. Nandimath, A. Patil, E. Banerjee, P. Kakade, S. Vaidya, “Big Data Analysis Using Apache Hadoop,” in IEEE IRI 2013, San Francisco, California, USA, pp. 14-16, Aug. 2013.

[25] J. Kestelyn, Putting Spark to Use: Fast In-Memory Computing for Your Big Data Applications, Cloudera Engineering Blog, 2013

[26] “Apache Spark MLlib,” http://spark.apache.org/mllib/, Retrieved 04-03-2017.

★★★