ISSN: 2278-4632
Vol-11 Issue-02 2021

BIG DATA ANALYSIS USING APACHE SPARK MLLIB AND HADOOP HDFS WITH SCALA AND JAVA

Dr. G Samba Siva Ra, Dr. Nedunchezhian.T
[1](Computer Science & Engineering, Gandhi Engineering College, Bhubaneswar)
[2](Computer Science & Engineering, Gandhi Engineering College, Bhubaneswar)

Abstract:
Nowadays with the technology revolution the term of big data is a phenomenon of the decade moreover, it has a significant impact on our applied science trends. Exploring well big data tool is a necessary demand presently. Hadoop is a good big data analyzing technology, but it is slow because the Job result among each phase must be stored before the following phase is started as well as to the replication delays. Apache Spark is another tool that developed and established to be the real model for analyzing big data with its innovative processing framework inside the memory and high-level programming libraries for machine learning, efficient data treating and etc. In this paper, some comparisons are presented about the time performance evaluation among Scala and Java in apache spark MLlib. Many tests have been done in supervised and unsupervised machine learning methods with utilizing big datasets. However, loading the datasets from Hadoop HDFS as well as to the local disk to identify the pros and cons of each manner and discovering perfect reading or loading dataset situation to reach best execution style. The results showed that the performance of Scala about 10% to 20% is better than Java depending on the algorithm type. The aim of the study is to analyze big data with more suitable programming languages and as consequences gaining betterperformance.

Keywords: Big data, Data analysis, Apache Spark, Hadoop HDFS, Machine learning, Spark MLlib, Resilient DistributedDatasets(RDD).

I. INTRODUCTION

At the present time with the huge improvement in the information technology field, and the facilitation over the internet for the billions of people through a huge database with the diversity of digital devices, the term of "big data" came out as a result. To put in a nutshell big data is the announcement of the enormous dataset size [1]**.** In the year2020, the number of linked devices will be roughly one hundred billion thus**,** guiding to supplementary data aggregation. Consequently clarifying and understanding bigdataanalyticstechniquesarebeingessential[2].As well as to the needs of changing from the traditional database (Relational DB) that has many limitations with the big data into NoSQL database (Non-Relational DB) which is overcome these limitations and suits the enterprise requirements [3]. Big Data mainly has three features which are recognized by 3Vs (Volume, Variety and Velocity). Some additional establishments and big data experts ha**ve** expanded this 3Vs framework to 5Vs framework by adding the terms of Value and Veracity into the big data explanation as shown in the Figure1 and shortly reported as follows[4][5][6]:

1. Volume: denotes to big quantities of data from diverse palaces, for example, mobile data, computers, servers and etc. The advantage of treating and studying these great sizes of data is earning valuable information on society and enterprises.
2. Velocity: states the swiftness of transferring data. The contents of data are regularly varying through the data gatherings process and resulting in different forms which are from several sources. This viewpoint needs new procedures and techniques for sufficiently exploring the streamingdata.
3. Variety: mentions collecting different kinds of data through different devices such as videos, images, etc. Furthermore, these kinds of data might be unstructured, semi-structured andstructured.
4. Value: denotes to the manner of pulling meaningful knowledge from enormous datasets. Value is the greatest significant feature of any big data tools since it permits to producing beneficialinformation.
5. Veracity: refers to the knowledge exactness or accuracy (informative andvaluable).

ISSN: 2278-4632
Vol-11 Issue-02 2021

Figure 1: Big data features

The aims of this article are for acquiring some knowledge on analyzing big data through up to date tool which is apache spark and hiring a few programming languages that fully compatible with it. Also, the target is transforming from traditional data stores (local disk) to big data requirements like HDFS.

APACHEHADOOPANDAPACHE SPARK

A. ApacheHadoop:

Hadoop is an open-source structure written in Java programming language, it permits to analyse big data through the clients. Hadoop can scale up an individual host to thousands of hosts with providing storage and calculation for each one. Basically, the Hadoop framework separated into three parts which are: Map Reduce, Yarn and Hadoop Distributed File System (HDFS)[8] as shown in figure2.

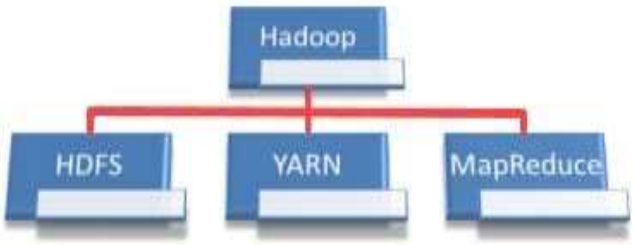


Figure 2: Hadoop Parts [8]**.**

MapReduce

Map Reduce part permits parallel processing of giant dataset. It changes a big portion of data into smaller ones to be treated individually on diverse data clients and automatically collects the outcomes through the several clients to bring back a sole result. Structured, semi, structured and unstructured data It can be processed. The structure consists of arrangement jobs, monitoring jobs and re-running all the failed job[9].

YARN

YARN stands for Yet Another Resource Negotiator and it works as a Hadoop cluster resource manager which means handling the Hadoop cluster resources such as Memory, CPU, etc. Fortunately, version 2 and 3 of Hadoop with Yarn opens a new door for data treating environment [10].

HDFS

Hadoop Distributed File System (HDFS) generally divides the file systems into data and metadata. HDFS has two important benefits in comparing with the traditional distributed file system. The first one is the great mistake tolerance because it saves the duplicates (copy) of the data in several data clients, which permits for distinguished error to recover data from other data clients. The second benefits it allows to use of big data sizes because the Hadoop clusters can residence data sets in petabytes[11].

B. ApacheSpark:

It is a model that performs a common data analysis on one node and distributed nodes which means it is similar to Hadoop. One of the advantages that it gives in memory calculations technique for increasing data processing speed. As well as, it can access Hadoop data storage (HDFS) because it runs on the top of the existing Hadoop node. Besides that, it can process the Streaming data like Twits on Twitter in addition to the structured data in Hive [12]. Basically, Spark divided into some parts and each part has its crucial task as shown in figure3.

ISSN: 2278-4632
Vol-11 Issue-02 2021

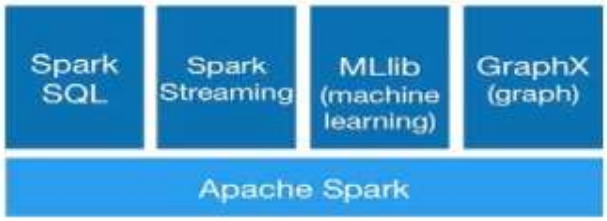


Figure 3: Apache spark parts

Spark Core is essential for the entire project. It offers distributed task, scheduling, and I/O jobs. Spark uses a particular data structure known as RDD (Resilient Distributed Datasets) that is a logical collection of data and separated over machines. RDD is Spark's primary abstraction, which is a fault-tolerant collection of elements that can be operated in parallel. They are immutable once you create an RDD. They can be transformed but they cannot be changed. They help to rearrange the computations and optimizing the data processing [12]**.**

Besides to Spark Core, a few further subparts like SQL, Streaming, Machine Learning library and Graph X are existing. All these parts have been established to complement the job of the core. The whole subparts constructed on the top of the core [13].

Programming languages inSpark

Spark consists of many language libraries that support performing various big data analysis. Spark written in Scala so, support it perfectly and even the startup of spark shell taking the users to Scala prompt automatically as shown in figure 4. In addition to Scala, three other programming languages exist in spark APIs which are Java, Python andR.

Since the structure of spark constructed in Scala, therefore writing a program using Scala language in spark offers to get the newest characteristics that may not exist in the further mentioned languages[14]. The sizes of Scala code are naturally smaller than the equivalent size of Java code. A lot of establishments that rely on Java in their works are changing to Scala to enhance the scalability and reliability [15].

Figure 4: Spark startups with Scala language.

Spark read (load) the data from everyplace

Spark can read or access the data that stored on Hadoop HDFS, Mesos, Mongo DB, Cassandra, H-Base, Amazon S3 and the data source from the cloud. Which means it has diversity access to data sources as shown in figure 5 and this is one of the advantages of spark[16].

ISSN: 2278-4632
Vol-11 Issue-02 2021

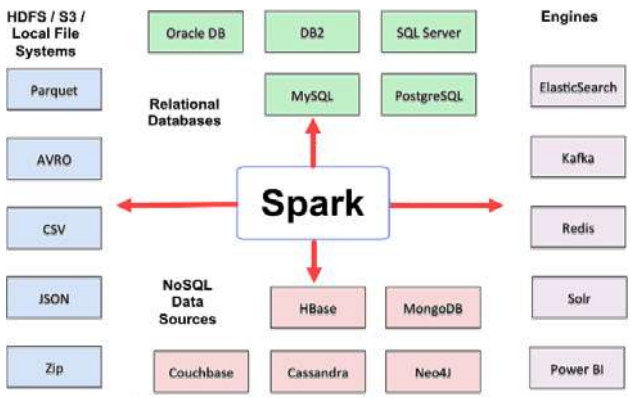


Figure 5: Spark diversity access to the data

Spark machine learninglibrary

In general, machine learning algorithms according to the style of the training data categorized into supervised and unsupervised learning. The machine learning of spark permits the data analytics and this library generally, contains the famous algorithms as shown in figure 6.

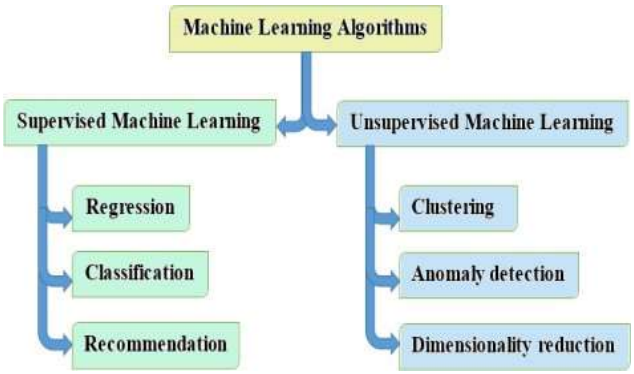
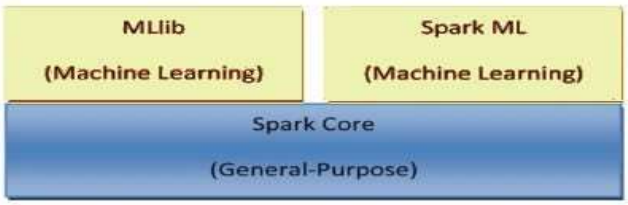


Figure 6: Machine learning categorization

Basically, the spark machine learning separated into two sets as shown in figure7. The first set is MLlib and it was built on the top of Resilient Distributed Datasets (RDD). It covers the common approaches that proposed so far. The next set is ML and it originates with the newest structures of MLlib for building ML pipelines and this Application Program Inter phase (API) is constructed on the Data Frames features [18].in this paper, the focus will be on the first set which is



MLlib.

Figure 7: Spark machine learning bundles

ISSN: 2278-4632
Vol-11 Issue-02 2021

II. RELATED WORK

A huge number of articles are printed on the subject of Apache spark recently. This article is the extension of the previous research papers published in this field. A lot of the researcher execute different algorithms utilizing spark and provide much great work based on the model of spark. For instance, some of them applying the algorithms of spark MLlib bundle and the others applying the algorithms of spark ML bundle in the analysis procedure. There have been several approaches for analyzing big data with spark this part of the paper focus on the latest movements and contributions in the first mentionedfield.

H. Sayedand et al, 2018 [19] in their paper compared the hypothesis that the Spark ML bundle has preference over the Spark MLlib bundle in the issues of performance and accurateness when dealing with big data. They discovered that the MLlib is better in the training time operation and vice versa in the evaluation time operation.

In addition, S. Al-Saqqa and et al, 2018 [20] discussed on the Spark's MLlib for hiring it in the classification of sentiment big data scale. They find out that support vector machine (SVM) is better than the other classifiers in the matter of performance.

As well as, K. AL-barznji and et al, 2018 [21] talked about sentiment analysis utilizing the algorithms of machine learning such as Naïve Bayes and SVM for analyzing the text with benefits of the huge capabilities of Apache Spark. They found that the SVM is more accurate in the condition of totalaverage.

However, M. Assefi and et al, 2017 [22] explored some views for growing the form of the Apache Spark MLlib 2.0 as an open source, accessible and achieve many machine learning tests that related to the real world to inspect the attribute characteristics. Also presents a comparison among spark and Weka with proving the advantages of spark over the Weka in many sides like the performance and it is efficient dealing with a huge amount of data. on the other hand, Weka is good for simple users with its GUI and the diversity of algorithms which already exist init.

Also, S. Salloumand et al, 2016 [23] stated an assessment on the key structures of big data analytics using Apache Spark. Furthermore, concentrates on the portions, concepts and topographies of Apache Spark and displays the advantages of it in the machine learning, analysis of the graph and stream treating in the enormous data fields. However, exposed the spark APIs and its compatibility with various programming languages in addition to the characteristics of the spark (RDD and data frame).

Likewise, A. Shoroand et al, 2015 [24] discovered some Big Data Analysis thought and distinguished a few important evidence from various big data streaming sources like twits of Twitter with applying Spark tools on it.

Moreover, A. Bansod 2015 [25] This researcher provides a newer rating work with storing a huge Dataset in the Hadoop Distributed File System HDFS and then analyzing it by Apache Spark. Also, present a comparison among spark and Hadoop Map-Reduce with showing the preference of the first one in performance and scalability.

Besides that, S. N Omkar and et al, 2015 [26] they applied a variety of classification methods on various datasets from the Repository of Machine Learning (UCI). As well the execution time and the accuracy of every classifier is discovered with some comparisons betweenthem.

Similarly, S. Gopalani and et al 2015 [27] compared the Hadoop Map Reduce and the Apache Spark and then provide a brief analysis of their performance by applying the K-Means algorithm.

In this paper, the focus will be on the imaginative and valuable ways of spark MLlib package that applied in big data study in the present time, by mentioning updating founded weakness and powerfulness with presenting the basic advantages and disadvantages and also showing the performance for the most famous machine learning algorithms with Java and Scala.

III. METHODOLOGY

In this paper, two types of programming language have been utilized, the first one is a Java programming language and the second one is Scala programming language. Both types evaluated in 32-bits and 64-bits Linux operating system environment because the Windows operating system is not efficient for processing big datasets and also not support big data tools like Hadoop. For both languages two different machine learning algorithms have been used, one of them is supervised machine learning which is Decision Tree Regression algorithm and the other one is unsupervised machine learning which is Clustering algorithm.

ISSN: 2278-4632
Vol-11 Issue-02 2021

Each algorithm read the dataset two times which means from two different places, one time the algorithm read the dataset that stored previously in the local hard disk drive and the second time read the dataset that stored or uploaded previously to the Hadoop HDFS storage. In summary 16 tests have been done, 8 tests for Java and the same for Scala. 4 java tests in 32-bits Linux OS and the other 4 Java tests in 64-bits Linux OS. Also, the same tests applied for Scala as shown infigure8.

Figure 8: Tests structure

TestedEnvironments

Two VMWARE environment have been utilized to gain these experimental outcomes. The first one is installed on 32-bits O.S and the second one installed on 64-bits. The rest information about used environments shown in table 1.

Table 1. Tested Environments.

No.	Resource Type	Details
1	Host O.S	Windows 10, 64-bits
2	Guest O.S	Debian 8 32bits Debian 9 64bits
3	VMware Version	15.0.2 build-10952284
4	CPU	Intel(R) Core(TM)i5-4300U CPU @ 1.90 GHz 2.50GHz
5	Number of VMware Processors	Four core
6	VMware RAM	8 GB
7	RAM in total	12 GB
8	HDD for VMware	40 GB
9	HDD in total	120 GB
10	Type of hard drive	SSD
11	Number of nodes	One node
12	Heap size	Default 1 GB
13	Hadoop Version	2.7.1
14	Spark Version	spark-2.4.0-bin-hadoop2.7

Datasets

Two different big datasets have been utilized in the Apache Spark MLlib 2.0 for analyzation process. Both of the data are from the Machine Learning Repository (UCI). The first one called Year Prediction MSD and it is about the estimation of the publishing year of a song from audio characteristics. The second dataset holds five groups of text in the formula of bags of words and we took only docword.nytimes from it [28]. Table 2 shown additional details on thedatasets.

Table 2: Characteristics of the dataset

Name of the Dataset	Type of Algorithm	Size	Characteristics	Attribute Characteristics
YearPredictionMSD	Regression	566 MB	Multivariate	Real
Bag of Words	Clustering	1.1 GB	Text changed to Libsvm	Integer

ISSN: 2278-4632
Vol-11 Issue-02 2021

IV. ANALYSIS, EVALUATION, AND FINAL EXPERIMENTAL RESULTS

This section is concisely demonstrating all the sixteen tests in four experimental scenarios, hence each scenario exhibits four tests as shown in Table 3 and described in following.

Table3: Duration of processing time for all Tests.

			Scala Min:Sec		**Java** Min:Sec	
OS Linux	**Data Size**	**Algor. type**	**Proc-time (Disk)**	**Proc-time (Hadoop)**	**Proc-time (Disk)**	**Proc-time (Hadoop)**
32-bits	1.1 GB	K-means Clustering	28.35	31.14	36.39	45.50
32-bits	566 MB	DTR	01.27	01.32	01.48	02.07
64-bits	1.1 GB	K-means Clustering	25.58	30.11	30.47	43.33
64-bits	566 MB	DTR	0.41	0.59	1.05	1.46

In scenario one, clustering K-means algorithm was applied on 1.1 GB dataset in the Linux 32-bits O.S environment. Two tests with Java have been done, one of them the data was loaded from local disk and the other the data was loaded from the HDFS. Similarly, two tests for Scala have been done, one test the data was loaded from the local disk and the other the data was loaded from HDFS.

What concluded from scenario one as shown in figure9 that the Scala programming language is faster than java when we use a spark MLlib clustering algorithm. However, reading the data from the local disk is faster than reading from theHDFS.

In the second scenario, the decision Tree Regression algorithm was applied on 566MB dataset from the type of LIBSVM in the Linux 32-bits O.S environment. Two tests with Java have been done, one of them the data was loaded from the local disk and the other the data was loaded from the Hadoop (HDFS). Similarly, the same two tests were applied with the Scala. What concluded from this scenario as shown in figure10 that the Scala programming language is faster than Java programming when we use a spark MLlib Regression algorithm. However, reading the data from the local disk is faster than reading from the HDFS in both programming languages.

In the third scenario, the same processes of the scenario one with the same dataset and algorithm have been repeated the sole difference is the Linux O.S environment changed to 64-bits. What concluded besides to what observed and stated from the scenario one that if we use much bigger dataset the OS 32-bits environment may not deal with it due to the heap size space problem because the higher heap size for the mention OS is close to 2 GB. More illustrations about the duration time of this scenario in figure11.

In the final scenario, the same processes of scenario two with the same dataset and algorithm have been repeated also the sole difference is the Linux O.S environment changed to 64 bits. What concluded besides to what observed and stated in the (Second and Third scenarios)

that if the dataset contains a huge number of attributes or if we put a massive number of Depth and Bins in any supervised algorithms similarly, the heap size space problem or garbage collection problem will appear in the 32-bits OS and it can't be solved due to the mentioned reason. Figure 12 showed the time difference between every test in this scenario.

V. DISCUSSION

It is good to state some problems during the test processes with presenting the solutions too:

ISSN: 2278-4632
Vol-11 Issue-02 2021

- **Not enough space in the temp:** Because the operation of the processing totally being located in the temp folder of the Linux 32-bits O.S normally the size of a temp folder is in MB and cannot stand big data size. The solution is increasing the size of temp into 1 GB from the Terminal as root by below command for affording huge datacalculation.
- **Java heap size space is not enough:** at the beginning of using any IDE (Integrated development environments) like Eclipse, NetBeans or any others IDE, the default heap size space for the project is between 64 to 256 MB and that space is not enough for computation large dataset. The solution is to increase it into 1GB to be like the spark default heap space. In bellow paths of theIDE:

Click on application Name – Properties-Run-VM Option- then set 1GB.

- **Weka can't read a huge data:** in the beginning, our intent was to compare the time performance between what presents in this paper and the normal Weka program but unfortunately, Weka cannot afford a huge dataset file. Especially it needs much more time than the spark just for reading without processing it. The solution is to change the Weka environment and adding a new package like distributed Weka Hadoop or spark and that might be a good topic to furtherresearch.

- **Windows O.S can't deal with big data:** windows can support Apache spark but it has a problem when processing a huge dataset in addition, it does not support Apache Hadoop which means Hadoop doesn't set up on windows environment. So the solution is utilizing the Linux operatingsystem.

VI. CONCLUSION

In the era of Big Data that we live in it, many new analytic tools like Apache Spark have been established to treat the procedures of Big Data perfectly. Spark offers a great performance and fault tolerant model with the scalability in analyzing big data framework. This paper works on the Apache Spark for big data analysis and then compares the Scala of spark with the Java in the spark MLlib bundle. It is seen that the Scala of Spark raises the speed calculation of the algorithms and finishes them in less time as compared to Java. This preference of Scala noticed in supervised machine learning algorithms such as Regression and unsupervised machine learning algorithm like Clustering. In addition, this research compared loading a big dataset from local disk and from the Hadoop HDFS storage. it appears that local disk is a little bit faster than HDFS but, it could be much faster if the Hadoop being distributed not in a single node like this paper. Because there are many elements that affect the time factor in the Hadoop distribution environment such as the method of connection. On the other hand, the advantage of the HDFS over the local disk is holding or storing Petabytes of data in case if it is being distributed which is the local disk absolutely cannot handle it. Further researches may find the optimal Big data analysis procedures and provide an efficient solution in thisfield.

The future work of this paper could be applied with four additional modifications. The primary one is changing the environment from a single node cluster into a multi-node and definitely that lead to gain better performance with the capability of executing larger data sets. The next alteration is reading the dataset from variety of storage that supports big data atmosphere for instance, Mongo DB, HBase, Cassandra, Couch-base and etc. for comparing which storage is more compatible with the spark and as consequences that decrease the request time which means reducing the execution time in the end. The third modification is comparing the other programming language performance in the matter of machine learning which is already supported by spark such as, R and Python. The final modification is using all the previous changes with the second bundle of spark machine learning library which is ML instead of spark MLlib bundle. Because ML bundle builds on the dataset and data frame, unlike the MLlib bundle that builds on the RDD. Then demonstrate the accuracy and the performance of each bundle with a comparison betweenthem.

REFERENCES

[1] N. Ibrahim, Alaa Hassan, Marwah Nihad, "Big data analysis of web data Extraction," International Journal of Engineering & Technology, vol. 7 (4.37), p. 168,2018.

[2] "Government Office for Science. The Internet of Things: making. 2014. The Internet of Things: making," 2014. [Online]. Available: https://www.gov.uk/government/uploads/system/uploads/attachme nt_data/file/409774/14-1230-internet-of-things-review.pdf. [Accessed 1 32019].

ISSN: 2278-4632
Vol-11 Issue-02 2021

[3] K. Jumaa, "Secured Data Conversion, Migration, Processing and Retrieval between SQL Database and NoSQL Big Data," DIYALA Journal for pure sciences, vol. 14, no. 4, p. 68, October 2018.
[4] G. Bello-Orgaz, J. J. Jung, D. Camacho, "Social big data: Recent achievements and new challenges," Elsevier B.V Inf. Fusion, Vols. 28,, p. 45–59,2016.
[5] K. AL-BARZNJI, A. ATANASSOV, "A SURVEY OF BIG DATA MINING: CHALLENGES AND TECHNIQUES," in Proceedings of 24th International Symposium "Control of Energy, Industrial and Ecological Systems, Bankia, Bulgaria, May2016.
[6] R. JANOŠCOVÁ, "Mining Big Data in WEKA," in 11th IWKM, Bratislava, Slovakia, October 20 – 21,2016.
[7] H. Özkösea, Emin Ari, Cevriye Gencer, "Yesterday, Today and Tomorrow of Big Data," Procedia - Social and Behavioral Sciences, vol. 195, p. 1043,2015.
[8] "Apache Hadoop," Apache software foundation, [Online]. Available: https://hadoop.apache.org/. [Accessed 6 32019].
[9] N. Pandey, Rajeshwari S, Shobha Rani BN, Mrs. Mounica B, "A Comparison on Hadoop and Spark," International Journal of Innovative Research in Computer and Communication Engineering, vol. 6, no. 3, p. 2062, March2018.
[10] Y. Perwej, Bedine Kerim, Mohmed Sirelkhtem, Osama E. Sheta, "An Empirical Exploration of the Yarn in Big Data," International Journal of Applied Information Systems (IJAIS), vol. 12, p. 19, December2017.
[11] T. Ruzgas, Kristina Jakubėlienė, Aistė Buivytė, "Big Data Mining and Knowledge Discovery," Journal of Communications Technology, Electronics and Computer Science, no. 9, p. 7,2016.
[12] Apache software foundation, 24 Feb 2019. [Online]. Available: http://spark.apache.org/.
[13] Diego García-Gil, Sergio Ramírez-Gallego, Salvador García, Francisco Herrera, "A comparison on scalability for batch big data processing on Apache Spark and Apache Flink," Big Data Analytics, p. 3,2017.
[14] Firoj Parwej,NikhatAkhtar,Dr. Yusuf Perwej, "A Close-Up View About Spark in Big Data Jurisdiction," V. Surekha. Int. Journal of Engineering Research and Application www.ijera.com, vol. 8, no. 1, p. 31, January2018.
[15] Tarun Kumawat, Pradeep Kumar Sharma, Deepak Verma, Komal Joshi, Vijeta Kumawat, "Implementation of Spark Cluster Technique with Scala," International Journal of Scientific and Research Publications, vol. 2, no. 11, p. 501, November2012.
[16] U. R. Pol, "Big Data Analysis: Comparision of Hadoop MapReduce and Apache," IJESC, vol. 6, no. 6, p. 6390,2016.
[17] B. Kaluža, Machine Learning in Java, UK: Packt Publishing Ltd, 2016.
[18] Salvador García*, Sergio Ramírez-Gallego, Julián Luengo, José Manuel Benítez, Francisco Herrera, "Big data preprocessing: methods and prospects," Big Data Analytics, p. 9,2016.
[19] Hend Sayed, Manal A. Abdel-Fattah, Sherif Kholief, "Predicting Potential Banking Customer Churn using Apache Spark ML and MLlib Packages: A Comparative Study," (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 9, pp. 674-677, Nov2018.
[20] Samar Al-Saqqaa, b, Ghazi Al-Naymata, Arafat Awajan, "A Large-Scale Sentiment Data Classification for Online Reviews Under Apache Spark," in The 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks, EUSPN Belgium,2018.
[21] Kamal Al-Barznji, Atanas Atanassov, "Big Data Sentiment Analysis Using Machine Learning Algorithms," in Proceedings of 26th International Symposium "Control of Energy, Industrial and Ecological Systems, Bankia, Bulgaria, May2018.
[22] Mehdi Assefi, Ehsun Behravesh, Guangchi Liu, and Ahmad P. Tafti, "Big Data Machine Learning using Apache Spark," in 2017 IEEE International Conference on Big Data, Boston, MA, USA, 11-14 Dec.2017.
[23] Salman Salloum, Ruslan Dautov, Xiaojun Chen1, Patrick Xiaogang Peng, Joshua Zhexue Huang, "Big data analytics on Apache Spark," Int J Data Sci Anal -Springer International Publishing Switzerland, September2016.
[24] Abdul Ghaffar Shoro, Tariq Rahim Soomro, "Big Data Analysis: Ap Spark Perspective," Global Journal of Computer Science and Technology: C Software & Data Engineering, vol. 15, no. 1, pp. 7- 14,2015.
[25] Bansod, "Efficient Big Data Analysis with Apache Spark in HDFS," International Journal of Engineering and Advanced Technology (IJEAT), vol. 4, no. 6, pp. 313-315, August 2015.
[26] Mohit1, Rohit Ranjan Verma, Sameeksha Katoch, Ashoka Vanjare, S N Omkar5, "Classification of Complex UCI Datasets Using Machine Learning Algorithms Using Hadoop," International Journal of Computer Science and Software Engineering (IJCSSE), vol. 4, no. 7, pp. 190-198, July2015.
[27] Satish Gopalani, Rohan Arora, "Comparing Apache Spark and Map Reduce with Performance Analysis using K-Means," International Journal of Computer Applications (0975 – 8887), vol. 113, pp. 8-11, March2015.
[28] "UCI machine learning repository," [Online]. Available: http://archive.ics.uci.edu/ml/index.html. [Accessed 26 22019].