

Random forest for lung cancer analysis using Apache Mahout and Hadoop based on software defined networking

Ali Abbood Khaleel¹, Ahmed Adnan Mohammed Al-Azzawi², Aws Mohammed Alkhazraji¹

¹Department of Computer Engineering Techniques, Bilad Alrafidain University College, Baqubah, Irak

²Department of Administrative and Financial Affairs, University of Diyala, Diyala, Iraq

Article Info

Article history:

Received Apr 13, 2023

Revised Aug 2, 2023

Accepted Aug 10, 2023

Keywords:

Apache Mahout

Big data

Hadoop

Machine learning

MapReduce

Random forest algorithm

Software defined networking

ABSTRACT

Random forest is a machine learning algorithm that mainly built as a classification method to make predictions based on decision trees. Many machine learning approaches used random forest to perform deep analysis on different cancer diseases to understand their complex characteristics and behaviour. However, due to massive and complex data generated from such diseases, it has become difficult to run random forest using single machine. Therefore, advanced tools are highly required to run random forest to analyse such massive data. In this paper, random forest algorithm using Apache Mahout and Hadoop based software defined networking (SDN) are used to conduct the prediction and analysis on large lung cancer datasets. Several experiments are conducted to evaluate the proposed system. Experiments are conducted using nine virtual nodes. Experiments show that the implementation of random forest algorithm using the proposed work outperforms its implementation in traditional environment with regard to the execution time. Comparison between the proposed system using Hadoop based SDN and Hadoop only is performed. Results show that random forest using Hadoop based SDN has less execution time than when using Hadoop only. Furthermore, experiments reveal that the performance of implemented system achieved more efficiency regarding execution time, accuracy and reliability.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Ali Abbood Khaleel

Department of Computer Engineering Techniques, Bilad Alrafidain University College

Baqubah, Diyala, 32001, Iraq

Email: alikh861@gmail.com, draliak@bauc14.edu.iq

1. INTRODUCTION

The internet and communication development leads to emerge the internet of things (IoT) technology where each thing, object and device is connected to the internet [1]. However, massive amounts of data are generated due to the communication between IoT devices and the internet. Furthermore, massive data are also produced from variety of sources including social websites, healthcare system, banking, educational platforms and industry. The storage and processing of such data has become difficult with traditional tools due to the characteristics of these data. Such data has different characteristics like volume, variety, velocity, veracity and value [2]. As a consequence, many companies developed some robust storage, processing and analysis frameworks to handle such data. The analysis and querying of such large and complex data to gain insights through traditional tools is a arduous task. To address this issue, many platforms have been built. Hadoop MapReduce platform is one of many solutions. Hadoop MapReduce is an open-source platform which provides distributed analysis and processing for massive and complex data on account of its scalability, reliability and fault tolerance. Hadoop is a framework that designed to allow processing of large amount of datasets on multiple computing nodes using the MapReduce algorithm [3], [4]. Additionally, enormous amounts of data

are processed and stored through many physical or virtual machines on top of Hadoop framework to obtain the maximum usage of storage and processing resources. Many machine learning techniques and data mining approaches have been built to analyze, mine and process big datasets efficiently. This paper implemented scalable machine learning using Apache Mahout based on Hadoop cluster and software defined networking (SDN). Random forest is employed as a classification method using Apache Mahout in Hadoop environment to perform the analysis on lung cancer datasets. The organization of the paper is shown as follows. Section 2, shows the most related work. The overview of Hadoop architecture, MapReduce and Hadoop distributed file system (HDFS) is given in section 3. Section 4 presents the applications of machine learning using Hadoop framework and software defined networking. Moreover, section 5, explains the proposed system. while section 6 presents the results and discussions and eventually, section 7 concludes the paper.

2. RELATED WORK

Several research works and projects have suggested and implemented Apache Mahout as a scalable machine learning system to provide high performance and computational analysis. Hadoop and Mahout in [5] have been used to implement random forest algorithm in a distributed environment. Four machines in Hadoop cluster have been employed to run the forest random algorithm. The performance analysis has been measured by analyzing the execution time, accuracy, kappa, standard deviation and reliability. Makeswar *et al.* [6] has employed Apache Mahout and Apache Flume as a scalable system based on Hadoop framework for large scale analyses of sensor network for air pollution observation over a large geographical area. The proposed work has utilized Hadoop distributed file system to store huge amount of sensor data and perform the analysis using Hadoop cluster and Apache Mahout. Apache Mahout is used by [7] to build a generic product recommender based on user based collaborative filtering to assist businesses to drive the sales of their products and gain more revenue. Srinivasulu *et al.* [8], Apache Mahout is employed to manage high dimensional datasets on top of Hadoop cluster by performing data fetching, text mining, clustering, classification and collaborative filtering. Furthermore, Apache Mahout is used to overcome different cluster problems such as cluster tendency, partitioning, cluster validity, and cluster performance. Another work used Apache Mahout as a cloud computing framework to compare k-means and fuzzy c-means for clustering a noisy realistic and big dataset for Wikipedia's latest articles [9]. Daoping *et al.* [10] used Apache Mahout application programming interface (API) to implement parallel k-means clustering algorithm to accelerate clustering for large-scale datasets.

The performance of Apache Mahout for clustering algorithms is analyzed by [11] using datasets of twitter streaming across several nodes of Hadoop MapReduce cluster under different evaluation criteria. Apache Mahout is used to provide a novel approach for collaborative filtering recommender system. The proposed work aims to address the concerns of sparsity and scalability by using dimensionality reduction method of linear algebra and ontology method for semantic similarity [12]. Bhatia *et al.* [13] employed Apache Mahout as a scalable data mining library based on Hadoop cluster to build a distributed generic recommender in order to produce recommendations about various items. Adekanbmi *et al.* [14], Apache Mahout is applied to train different classifiers of machine learning, such as random forest, logistic regression (LR), and Naive Bayes (NB). Apache Mahout is used to build an effective recommender system based on collaborative filtering using Eclipse on a sample dataset [15]. Deepak *et al.* [16], Apache Mahout, MapReduce and collaborative filtering are employed to provide efficient performance analysis on big data. Chitra and Jayalakshmi [17] analyze the medical threshold for chronic kidney diseases and cardiovascular diseases using internet of things. They used Hadoop, HBase and Apache Mahout to manage the large amounts of IoT sensor information and "aid in the development of its primary breaking religion figures models for chronic kidney diseases (CKD) and cardiac diseases".

3. HADOOP ARCHITECTURE

Hadoop is scalable framework proposed by Apache to enable parallel and distributed processing of enormous datasets through the MapReduce model. Hadoop is designed as a master/slave architecture that incorporates two parts, one for processing purposes called MapReduce and the other for storage called Hadoop Distributed File System. It consists of single master machine and multiple slave machines. The master machine uses job tracker to manage and monitor both of Map and Reduce tasks progress. on the other side, the slaves employ task trackers to run all tasks of MapReduce [18]. The following sections illustrate Hadoop distributed file system and MapReduce.

3.1. Hadoop distributed file system

HDFS is the main part of Hadoop framework that divides the storage of enormous data across multiple machines. HDFS is built in java in the same way of the file system, which was developed by Google to enable the distributed storage in Hadoop cluster. High throughput access to data applications can be provided due to

the availability and fault tolerance features [19]. HDFS takes the input data and divide it into many blocks and create many copies of them. These copies are distributed to be stored on several slave machines of the Hadoop cluster. The size of each data block in Hadoop is given by the user during cluster settings. Furthermore, data replicas are also set during the cluster setup. Each data block size is configured in the cluster by default 64 MB and can be increased to be 128 MB depending on the resources of Hadoop cluster. On the other hand, the replication factor is set to be 3 for each data block to ensure the availability of data in case of any error or failure on any machine in the cluster. As aforementioned, in HDFS, the duty of name node is to manage file system namespace. Moreover, the name node can perform many tasks like the creating files and directories in addition to some file operations. Name node has an alternative node called secondary name node which is used as a second name node to achieve high cluster performance. The instructions are sent by the name node to data nodes, which store the original data on the slave machines. The operations in regard to creating, deleting and replicating data blocks are performed by the data nodes [20].

3.2. MapReduce model

MapReduce is a system that mainly built to work with the Hadoop framework in order to facilitate the processing of huge data using single or multiple clusters in distributed environment. MapReduce programming model comprises two functions, namely Map and Reduce. These two functions are used to process datasets. These datasets are divided by the MapReduce job into multiple blocks. The data blocks are processed through three phases, the first one is the Map phase that takes the datasets to process them and transfer the results by the shuffling phase (second phase) to the third phase, which are the reduce phase to take out the final outcomes [4], [20]. MapReduce and Hadoop are widely adopted for many big data applications and machine learning algorithms.

4. MACHINE LEARNING ALGORITHMS IN HADOOP ECOSYSTEM

Enormous data sets are created by a variety of sources and applications every day. Such data requires mining tools to gain insights and obtain useful information to enhance the quality of our daily life. Machine learning [21] algorithms can enable computers to extract and predict useful information from data without any explicit programming. On the other hand, make prediction and perform extraction on large datasets is a crucial task, as it demands high-quality resources that are not easily provided by single device. Consequently, many tools have been designed to integrate the algorithms and techniques of machine learning with Hadoop clusters that facilitate the implementation of multiple machine learning algorithms. An introduction of Apache Mahout in Hadoop environment is described in the following section.

4.1. Apache Mahout

Apache Mahout is a distributed framework that was designed to enable data scientists, mathematicians and statisticians to implement their machine learning algorithms using Hadoop cluster. It is used to support scalable machine learning algorithms across multiple machines in a Hadoop cluster. Mahout can enable various techniques and approaches of machine learning including clustering, classification and recommendation [22]. It is a proper choice to implement a variety of applications in regard to machine learning like data mining, forecasting and pattern recognition. Mahout scales the machine learning algorithms by using the library of Hadoop. Many classification algorithms, such as random forest, hidden Markov model (HMM) and Naïve Bayes can be implemented based on Apache Mahout using Hadoop framework [23]. In this work, random forest will be employed as a scalable machine learning based on Mahout and Hadoop with SDN.

4.2. Random forest algorithm

Random forest is a machine learning algorithm that was proposed by Leo Breiman and Adele Cutler in 2001. It is considered as statistical learning model and classification algorithm that can learn based on decision tree. Random forest employs ensemble learning method for classification and regression. It builds several decision trees at training time and also obtains the class, which is the mode of the classes obtained through individual trees. Random forest combines many decision trees to obtain a single outcome, where each decision tree is created by dividing datasets into partitions and branches by using iterative process [24], [25]. Random forest algorithm has a better performance for traditional data mining duties. However, its performance decreases for big data applications because it builds multi-level decision trees that needs significant amount of memory and computing power. As a result, Hadoop cluster and Apache Mahout can be used to improve the performance of random forest. Section 5 shows the implementation of random forest algorithm using Mahout and Hadoop based SDN.

4.3. Software defined networking

Software defined networking known as SDN is a relatively new technique that mainly proposed to solve many networking problems in the traditional networks. SDN segregates control plane to be implemented separately from data plane to address limitations in the architectures of traditional networks. It provides centralized management, intelligent programming and agile network configurations. Both forwarding and routing procedures are carried out by the same network device in the current traditional networks. On the other side, when using SDN technology, the control plane is performed separately in a software layer, namely the SDN controller. The control plane is called SDN brain because it is used to perform the routing of packets and offers intuitive programming and centralized management. While, the data plane is responsible to perform the packet forwarding in the SDN network. Software defined network has three layers, which are application layer, control layer and infrastructure layer [26], [27].

5. THE IMPLEMENTATION OF SCALABLE RANDOM FOREST USING APACHE MAHOUT AND HADOOP BASED SOFTWARE DEFINED NETWORKING

Random forest algorithm is implemented based on Apache Mahout using Hadoop framework and SDN. The setup of random forest based on Apache Mahout in Hadoop cluster is configured according to [23]. In this paper, Apache Mahout is employed to provide easy implementation for random forest algorithm to analyze lung cancer datasets. Mainly, datasets are taken from Baqubah teaching hospital in Baqubah city. These datasets are classified as a big data due to its multidimension characteristics as well as the high number of instances. Furthermore, the number of instances will be duplicated randomly by random forest algorithm. The proposed system uses random forest algorithm as analysis and classification method to predict the lung cancer among men. The MapReduce algorithm and Hadoop are used to perform the processing across several machines in parallel. Furthermore, HDFS is used for storage purposes. The lung cancer datasets are pre-processed before it has been moved to HDFS. After the pre-processing stage, datasets are divided into multiple training and testing sets and moved to HDFS using the following commands, which are `Hadoop fs -mkdir /user/hue/LungCancerTrain` and `Hadoop fs -mkdir /user/hue/LungCancerTest`, respectively. Once datasets are loaded into Hadoop cluster, the analysis of lung cancer data is performed by random forest algorithm using Apache Mahout. On the other side, the processing is performed by MapReduce across Hadoop cluster with nine virtual machines. In this work, we used 30% of data as a training while the rest was for testing to provide appropriate ratio according to [28]. Descriptor file is created to label all training datasets information in order to understand which one is numerical and which one is categorical. The computation process is carried out on these data training and testing sets through multiple machines based on Hadoop cluster. The processing of data training and testing sets is performed by map tasks where each map task is responsible to process the assigned data. After the finishing of the map process, the reduce process begins to process the intermediate outcomes received from Map process. However, the shuffling phase, which is located between the Map and Reduce phase has a serious issue represented by the time taken to shuffle the results from the mappers to the reducers because it requires high bandwidth. As a consequence, SDN is employed to improve the shuffle phase during the MapReduce process. In our proposed work, we used SDN with OpenFlow switch to specify a proper bandwidth for the traffic produced during the shuffle stage when the mappers transfer their result to the reducers. The architecture design of our proposed system is illustrated in Figure 1.

5.1. The evaluation of the proposed system

Evaluation on lung cancer data analysis using random forest algorithm based on Apache Mahout in Hadoop cluster and SDN has been performed. Comparison between the implementation of random forest in traditional environment and its implementation using our proposed system based on Mahout and Hadoop with SDN has been carried out. The performance of random forest as a machine learning algorithm based on Apache Mahout and Hadoop framework using SDN is evaluated in terms of execution time, standard deviation (STD), reliability and classification accuracy. The evaluation on work performance was carried out on lung cancer datasets provided by Baqubah teaching hospital. These datasets have been duplicated to make them massive in order to simulate the scenario of big data. The proposed system is implemented by employing nine machines, one has been configured to be master node, while the rest are configured to be slaves. Oracle virtual box has been used to create the nine machines virtually. We assign Six Gigabyte of RAM and Two cores of CPU for each machine with 220 Gigabyte of storage for the whole cluster. Two high performance servers have been used to setup the cluster. Apache Hadoop (3.3.2) and Apache Mahout (14.1) and random forest algorithm are installed on Hadoop virtual nodes. SDN with two virtual OpenFlow switches are used to connect the two servers. The specifications of our system are shown below in Tables 1 and 2.

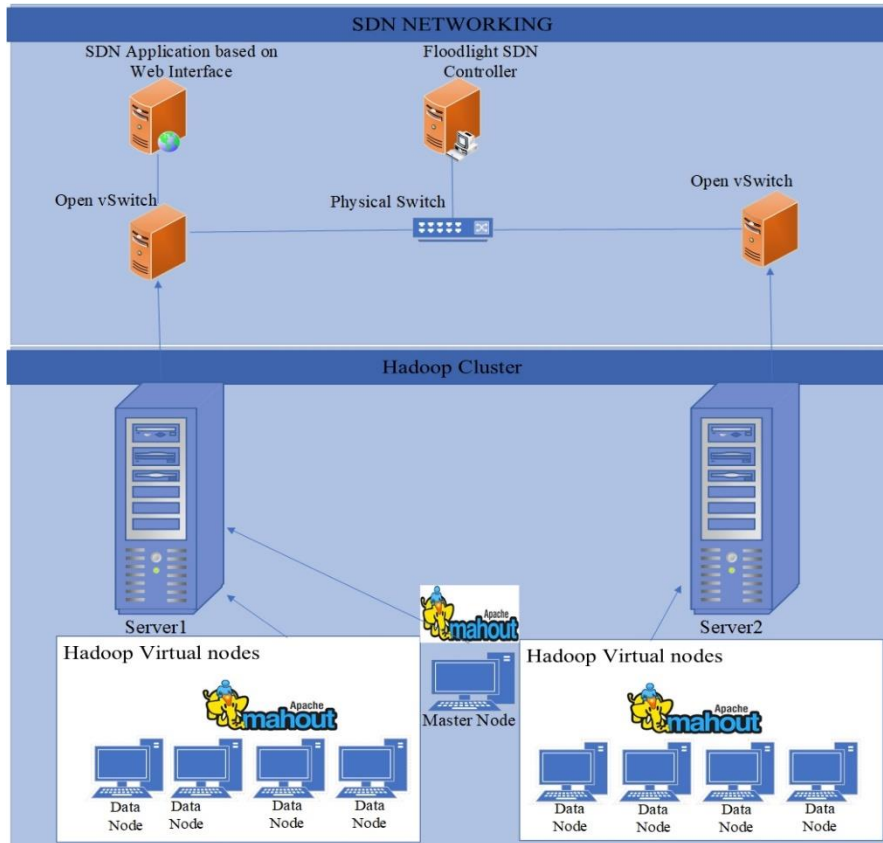


Figure 1. The architecture design of the proposed system

Table 1. Cluster specifications

Servers	Cluster resources	amount
"Intel Xeon X5550 server 1" and "uxisvm04 server 2"	"CPU"	"18 cores"
	"Processor"	"2.29 GHz"
	"Storage"	"220 GB"
	"Connection"	"1 Gbit Ethernet"
	"memory"	"54 GB"

Table 2. Software tools

Host operating system	Guest operating system	Software
"Microsoft windows server 2012 R2"	"Ubuntu Linux 18.04 (x86, 64-bit)"	Mahout 14.1, "Hadoop 3.3.2", OpenFlow Switch, SDN floodlight Controller

6. RESULTS AND DISCUSSION

Several experiments have been conducted to evaluate the performance of the implemented system. Various data sizes ranging from 100 instances to 20,000 instances have been used to evaluate the performance analysis of random forest algorithm in Hadoop cluster. The execution time of different instances has been evaluated to show the performance of random forest using the proposed work. Figure 2 reveals that the execution time of instances from 100 to 10,000 remains steady. This means that the use of Hadoop has a better efficiency to tackle such large datasets. Comparison between the implementation of random forest algorithm using single node in traditional environment and the implementation of random forest using nine machines of Hadoop cluster has been conducted. From Figure 3, it is noted that the execution time of random forest when using single machine in traditional environment is risen particularly for large data sizes. On the other hand, the execution time is reduced when using multiple machines in Hadoop environment because of the distributed processing of Apache Mahout through many machines in the cluster and, thus decrease the time by using more computing power and CPU of Hadoop machines.

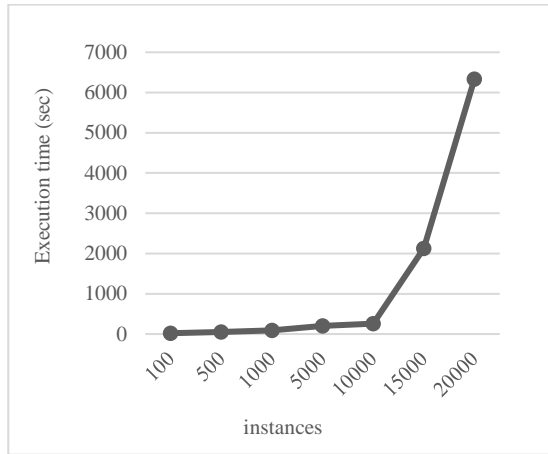


Figure 2. Execution time against different instances

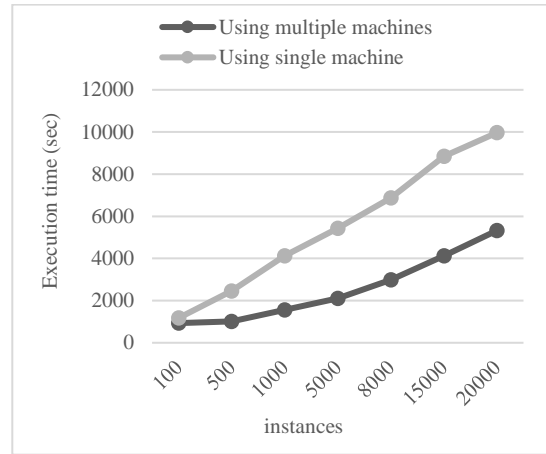


Figure 3. Using multiple machines vs single machine

A further comparison between the implementation of forest algorithm based on Apache Mahout in Hadoop cluster using SDN and without using SDN has also been performed. The performance of random forest implementation using our proposed work with SDN seems better than its implementation without using SDN. It is seen in Figure 4 that the time of execution using Hadoop with SDN is reduced comparing to using Hadoop only. This is because the SDN technology improved the performance of networking part of Hadoop cluster by reducing the shuffling time when moving the results from the mappers to the reducers during the process of Map and Reduce. In the SDN, the control plane is decoupled from the data plane, which leads to improve the networking side. Figure 5 reveals the classification accuracy of random forest using different percentages of training data. It is observed from Figure 5 that the accuracy of random forest classification is improved especially when the percentage of training data is increased.

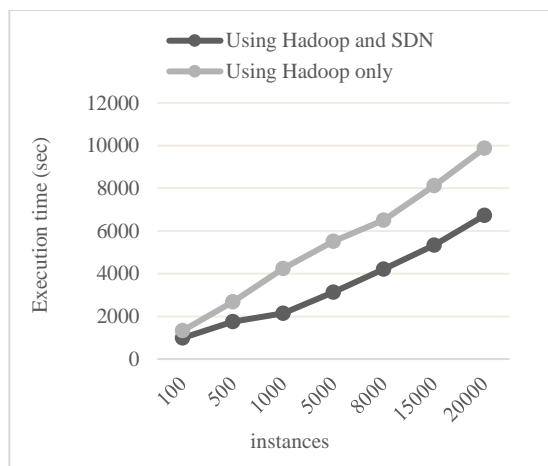


Figure 4. Using Hadoop and SDN vs Hadoop only

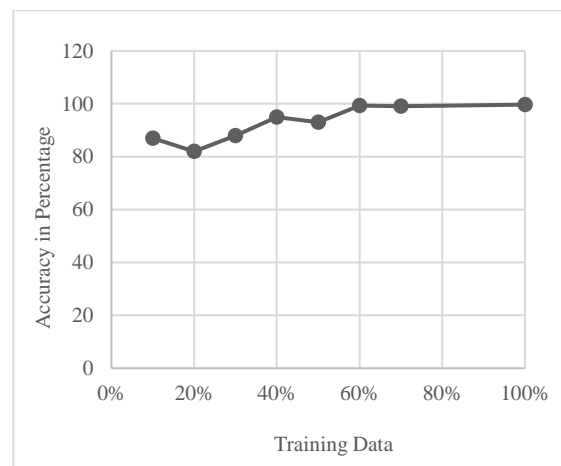


Figure 5. Accuracy of random forest

Standard deviation is used to measure the diffusion of data points. It is observed from Figure 6 that standard deviation keeps constant when data size rises. The average of data points is also measured to show the reliability when the size of data increases. It is seen from Figure 7 that the average of data points keeps constant when the data size increases. This shows that the use of Apache Mahout, Hadoop and SDN has a major influence on the performance of random forest algorithm due to the efficient utilization of computing and networking resources.

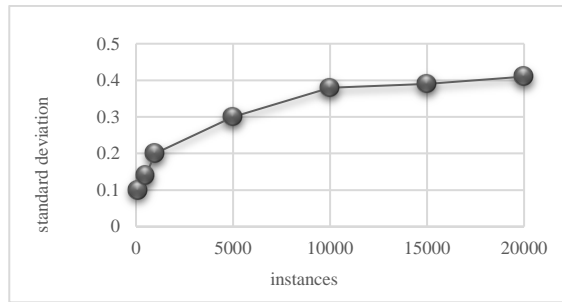


Figure 6. Standard deviation of random forest

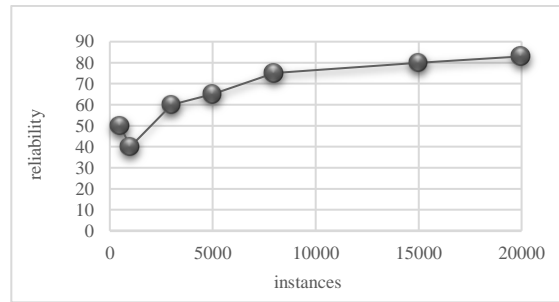


Figure 7. Reliability of random forest

7. CONCLUSION

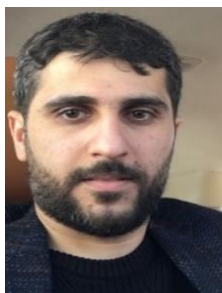
In this paper, random forest based on Apache Mahout has been run using Hadoop cluster and SDN to perform the analyses on lung cancer data. Comparison between the implementation of forest algorithm in traditional environment and the implementation of random forest using the proposed system was performed. Different experiments were conducted to evaluate the proposed system performance from different aspects. The evaluation on the performance of the proposed system has been performed through nine virtual machines of Hadoop cluster. Results reveal that the performance of random Forest implementation using our proposed work has better efficiency than its implementation in conventional environment. The results also reveals that our proposed work performance in terms of classification accuracy, standard deviation and reliability was improved by using SDN technology.




REFERENCES

- [1] Team tutorials point, "Internet of things (IoT) tutorial," *tutorialspoint*, 2017. https://www.tutorialspoint.com/internet_of_things/index.htm (accessed Apr. 09, 2023).
- [2] M. Naeem *et al.*, "Trends and future perspective challenges in big data," in *Smart Innovation, Systems and Technologies*, vol. 253, 2022, pp. 309–325.
- [3] "Apache Hadoop," *The Apache Software Foundation*, 2006. <https://hadoop.apache.org/> (accessed Apr. 09, 2023).
- [4] "MapReduce tutorial," *hadoop*, 2022. https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html (accessed Apr. 09, 2023).
- [5] K. Kumar, N. A. Sharma, and A. B. M. S. Ali, "Classification in a distributed system-A study of random forest in the Hadoop MapReduce framework," in *2019 IEEE Asia-Pacific Conference on Computer Science and Data Engineering, CSDE 2019*, Dec. 2019, pp. 1–6, doi: 10.1109/CSDE48274.2019.9162365.
- [6] P. B. Makeshwar, A. Kalra, N. S. Rajput, and K. P. Singh, "Computational scalability with Apache Flume and Mahout for large scale round the clock analysis of sensor network data," in *RAECE 2015 - Conference Proceedings, National Conference on Recent Advances in Electronics and Computer Engineering*, Feb. 2016, pp. 306–311, doi: 10.1109/RAECE.2015.7510212.
- [7] U. Farooque, "Implementing user based collaborative filtering to build a generic product recommender using Apache mahout," in *Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom 2016*, 2016, pp. 984–987.
- [8] A. Srinivasulu, C. D. V. Subbarao, and Y. J. Kumar, "High dimensional datasets using hadoop mahout machine learning algorithms," in *International Conference on Computing and Communication Technologies*, Dec. 2014, pp. 1–1, doi: 10.1109/ICCC2.2014.7066727.
- [9] C. Rong and R. M. Esteves, "Using Mahout for clustering Wikipedia's latest articles: A comparison between k-means and fuzzy c-means in the cloud," in *Proceedings - 2011 3rd IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2011*, Nov. 2011, pp. 565–569, doi: 10.1109/CloudCom.2011.86.
- [10] X. Daoping, Z. Alin, and L. Yubo, "A parallel clustering algorithm implementation based on Apache Mahout," in *2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, Jul. 2016, pp. 790–795, doi: 10.1109/IMCCC.2016.9.
- [11] F. Xhafa, A. Bogza, and S. Caballe, "Performance evaluation of Mahout clustering algorithms using a Twitter streaming dataset," in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, Mar. 2017, pp. 1019–1026, doi: 10.1109/AINA.2017.50.
- [12] D. Vats and A. V. Sharma, "A collaborative filtering recommender system using apache Mahout, ontology and dimensionality reduction technique," in *2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, Jan. 2022, pp. 1–12, doi: 10.1109/ACCAI53970.2022.9752604.
- [13] L. Bhatia and S. S. Prasad, "Building a distributed generic recommender using scalable data mining library," in *Proceedings - 2015 IEEE International Conference on Computational Intelligence and Communication Technology, CICT 2015*, Feb. 2015, pp. 98–102, doi: 10.1109/CICT.2015.129.
- [14] O. Adekanbmi, H. Wimmer, and J. Kim, "Big cyber security data analysis with Apache Mahou," in *2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications, SERA 2022*, May 2022, pp. 83–90, doi: 10.1109/SERA54885.2022.9806807.
- [15] Ruchika, A. V. Singh, and M. Sharma, "Building an effective recommender system using machine learning based framework," in *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, Dec. 2017, vol. 2018-Janua, pp. 215–219, doi: 10.1109/ICTUS.2017.8286008.




- [16] V. Deepak, M. R. Khanna, K. Dhanasekaran, P. G. O. Prakash, and D. V. Babu, "An efficient performance analysis using collaborative recommendation system on big data," in *Proceedings of the 5th International Conference on Trends in Electronics and Informatics, ICOEI 2021*, Jun. 2021, pp. 1386–1392, doi: 10.1109/ICOEI51242.2021.9452737.
- [17] S. Chitra and V. Jayalakshmi, "Analyze the medical threshold for chronic kidney diseases and Cardio Vascular diseases using internet of things," in *2021 4th International Conference on Computing and Communications Technologies (ICCT)*, Dec. 2021, pp. 189–193, doi: 10.1109/ICCT53315.2021.9711829.
- [18] T. White, *Hadoop: The definitive guide*. O'Reilly Media, Inc, 2009.
- [19] D. Borthakur, "HDFS architecture guide," *Hadoop Apache Project* http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/hdfs_design.pdf, 2008. http://archive.cloudera.com/cdh/3/hadoop-0.20.2-cdh3u6/hdfs_design.pdf%5Cpapers3://publication/uuid/BE03DF70-D0C1-441E-A65F-1888C84992D6 (accessed Apr. 09, 2023).
- [20] A. Khaleel and H. S. Al-Raweshidy, "Optimisation of hadoop MapReduce configuration parameter settings using genetic algorithms," in *Advances in Intelligent Systems and Computing*, vol. 857, 2019, pp. 40–52.
- [21] J. Hurwitz and D. Kirsch, "What is machine learning? | IBM," *Retrieved March*, 2018. <https://www.ibm.com/topics/machine-learning> (accessed Apr. 11, 2023).
- [22] "Apache Mahout," Mahout. The Apache Software Foundation, 2014, Accessed: Apr. 09, 2023. [Online]. Available: <https://mahout.apache.org/>.
- [23] A. Gupta, *Learning Apache Mahout classification*. Packt Publishing Ltd, 2015.
- [24] A. B. Shaik and S. Srinivasan, "A brief survey on random forest ensembles in classification model," in *Lecture Notes in Networks and Systems*, vol. 56, 2019, pp. 253–260, doi: 10.1007/978-981-13-2354-6_27.
- [25] IBM Cloud Education, "What is random forest? | IBM," *Ibm*, 2020. <https://www.ibm.com/cloud/learn/random-forest> (accessed Apr. 09, 2023).
- [26] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27–51, 2015, doi: 10.1109/COMST.2014.2330903.
- [27] CISCO, "Software-defined networking (SDN) definition - Cisco," *Cisco.com*, 2019. <https://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html#-what-is-sdn%0Ahttps://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html?dtid=ossdc000283%0Ahttps://www.cisco.com/c/en/us/solutions/software-defined-n> (accessed Apr. 09, 2023).
- [28] W. Buntine and T. Niblett, "A further comparison of splitting rules for decision-tree induction," *Machine Learning*, vol. 8, no. 1, pp. 75–85, 1992, doi: 10.1023/A:1022686419106.

BIOGRAPHIES OF AUTHORS






Ali Abbood Khaleel    received the Ph.D. degree in Electronic and Computer Engineering from Brunel University, London, U.K. in 2018. His research interests include big data analytics, cloud computing, high performance computing, parallel computing, distributed computing and storage, and software-defined networks. He can be contacted at email: draliak@bauc14.edu.iq.



Ahmed Adnan Mohammed Al-Azzawi    received his Master in Information Technology, from School of Natural and Applied Sciences, Cankaya University, Ankara, Turkey, in 2015, his research interests include computer network and communication, cloud computing and big data. He can be contacted at email: ad_ahmedal-azzawi@uodiyala.edu.iq.



Aws Mohammed Alkhazraji    received a bachelor's degree in computer Engineering from Diyala University, then a master's degree in electrical and computer Engineering from Altinbas University-Turkey in 2022. His research interests include Deep learning and computer networks. He can be contacted at email: awsalkhazraji@bauc14.edu.iq.