TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 1224 - 1231

Applicational Achievement of K-Means Algorithm among Apache Spark and Map Reduce

[1]Dr. E. Laxmi Lydia, [2]G Sandhya, [3]Hima BinduGogineni ,[4]Guvvu PavaniLatha[5]N.Sharmili
[1]Professor, Vignan's Institute of Information Technology (A), Department of Computer Science and Engineering, Visakhapatnam, Andhra Pradesh, India.elaxmi2002@yahoo.com
[2]Assistant professor,Vignan's institute of engineering for women
[3]Asstistant. Professor,Department of Computer Applications,Vignan's Institute of Information and Technology
[4]Assistant Professor,Dept of CSE,Vignan's institute of engineering for women
[5]Associate professor ,Computer science and engineering Department, GayatriVidyaParishad college of engineering for women, visakhapatnam, Andhra pradesh, India

Article Info
Volume 82
Page Number: 1224 - 1231
Publication Issue:
January-February 2020

Article History
Article Received: *14 March 2019*
Revised: *27 May 2019*
Accepted: *16 October 2019*
Publication: *06 January 2020*

Abstract
Tremendous data all around the globe have been an enthusiastic subject in computer science to explore and analyze that has raised the prominence of information. Blast incoming data through online networking,explorationin big organizations to get more access to intelligent research has become a great demand.MapReduce and its discrepancy have been very worthwhile in accomplishingenormouscalibratereports with robust applications on specialty groups. Therefore, a substantial quantity of the particular schemes is assembled over a non-cyclic intelligence flow and is not suitable to demonstrate for some other influential applications. An unbending architecture design was exclusively introduced using MapReduce that evaluates each job in a straightforward approach. Major steps in MapReduce such as a map, shuffle and reduce are allowed to change, synchronize and combine the outputs that are collected from every node cluster. Subsequently,to overwhelm the system to manual and recede, this paper proposes Apache Spark a manipulating form to split the tremendous information. The prime adversary for "successor to MapReduce" is Apache Spark. Similar to a broadly significant engine MapReduce, Spark has been designed to run distinctadditional workloads and to perform in that space witha greatlyaccelerated speedadapted framework. In this paper conflict between these two systems altogetherutilized with execution exploration by considering its information computation in a specified machine. Clustering process (K-Means) and asserting different criteria essentially, speed up the system, energy consumption of the system,scheduling delay of the jobthan the current systems.

Keywords*: Spark, MapReduce, Hadoop, BigData*

1. INTRODUCTION

Well, known cluster computing has broadly directed its process to data-parallel computations. These clusters are executed with uncertainty in systems that accordinglyproduce locality-receptiveprogram, detection of faults in components or any failures during execution, and distribution of loads through load balancing in clustering. MapReduceprompts this design, during machines like Dryad and data streams, are sorted after merging by MapReduce. The facilitator of Big Data [5] cloud computing[6] granted cloud storage in disturbed systems [9].

These systems accomplish their scalability and fault tolerance by giving a programming model where the client makes non-cyclic data stream graphs to go input data through an arrangement of operators. This permits the hidden framework to oversee scheduling and to respond to faults without client mediation. While this data flow-programming model is useful for a large class of applications, there are applications that cannot

January - February 2020
ISSN: 0193 - 4120 Page No. 1224 - 1231

communicate proficiently as non-cyclic data flows. In this paper, we concentrate on one such class of applications: those that reuse a working arrangement of data over numerous parallel operations [21-36]. This incorporates two use cases where we have seen Hadoop users report that MapReduce is lacking:

Iterative field: In this field, capacity is applied more than once to the same dataset using a lot of usual machine learning algorithms with inclination plummet to upgrade the parameter.Working withMapReduce and Dryadfor every iterationof communication a huge performance is attained as a drawback.

Intelligent logicalanalytics through the Hadoop ecosystem:Large datasets use SQL interfaces with exploratory questions run on Hadoop such as Pig [13] and Hive[11].In a flawlessglobe, any end-user may have the possibility of transferring the dataset into memoryfrom disparate machines and examine it more than once. Hadoop uses queries with Be that as it may, with Hadoop, every inquiry acquires huge inertness because it keeps running as a different MapReduce occupation and peruses data from disk.

In this paper, advanced computational spark helps to create a new cluster computational framework thatmaintains scalability and fault tolerance characteristics along with applicational working applications with working comparative settingsto MapReduce.

The underlying concept behind spark [19] is partitioned items are read quantitatively among large provision of systems for the reestablishment of lost segments. Memory expressly stores RDDinformation crosswise through clients over machines and change it in variousMapReduce-like coordinative procedures. RDDs manage fault tolerance over an intention of extraction; if any allowance of an RDD is missing, the RDD process adequacy detailsregarding it about how it was drawn from divergent RDDs to acquire the scope to transmute only that package. Although the experience that RDDs are neither a regularly shared memory consideration, they express to a sweet-spot at intervalsarticulation from one context,extensible and authenticity, and an appropriate mixture of applications were identified.

Data processing spark is implemented in Scala [15], written in high-level programming languagepassivelyconsidering the Java Virtual Machine, and DryadLINQfunctional programming interface is discovered. Besides, the spark can be handledrationally from a transformed translation of the scalaexponent, and also allows the client to represent RDDs, functional operations, volatile defined data and classify the data based on classes and apply them to correlate the operations on a cluster. This produces the accredit framework on spark concerning the appropriate process for large-scale datasets on a cluster.

Although the managing controlevidence,the spark is still a model that implementsempowering connections with the framework. It is observed that continual machine learning assignment Hadoop 10x was overtaken by a spark. and can be employedrationally to penetrate a 39 GB dataset including sub-second intermission.

1.1 HADOOP OVER SPARK

One of the most advanced data processing technology fora long time for Big Data [1]is Hadoop and is achieved by being the key to generate clarification onthe outcome for transforming substantial datasets. On the basis of one-pass MapReduce is an extraordinary solution for one-pass reckoning, yet not severely worthwhile for multi-pass reckoning and methods. Each stage in the data processing task mechanismincludes two phases such as map state and reducestate and based on the change over each utilization case into MapReduceorder to impact this outcome. The task outcome statisticsamong every betterment must be taken away in the distributedciteddesign before the process towards the step can start. Accordingly, this technique has a predisposition to be tolerablefor the reason of likeness. Furthermore, Hadoop solutions typicallyintegrategroups that are demanding to create and inspect. Additionally, itdemands the fusion of fewer machines for assorted big data [2]systems (a stream of data processing in machine learning such as Mahout).

January - February 2020
ISSN: 0193 - 4120 Page No. 1224 - 1231

Achievement of bringing out the required outcomes from complicated data, MapReduce [8]tasks are ordered together with development and implement performed tasks in series. The tasks that are allowed to execute have high-latency, and no new task is allowed to execute until the previous task is accomplished completely. Most complicated problems are resolved using Spark throughnon-cyclic diagram (DAG) design and multi-step data pipelines. Moreover, it depends on theangular data sharing in-memory overDAGs, remarkably disparate tasks can perform with identical data.

To gain improved and additional benefits in Spark [17], it allows tasks to run on the currently executed Hadoop distributed file system(HDFS) [15] .It delivers assistance to set up spark [18]operations in present Hadoop v1 chunk (that includes spark-inside-MapReduce (SIMR)) or rather Hadoop v2 YARN chunkon the contraryproportional open-source computer cluster such as apache mesos. A special attractive view at spark grabbed its attain alternatively toHadoop MapReduce [14] as conflicting a replacement to Hadoop. It neitherrecommendedto succeedHadoop but relativelywell-ranging bound well-adjusted responses for supervisingcharacteristic big data stretches out and essentiality. Figure 1 professed the oppositionthroughout hadoop and spark.

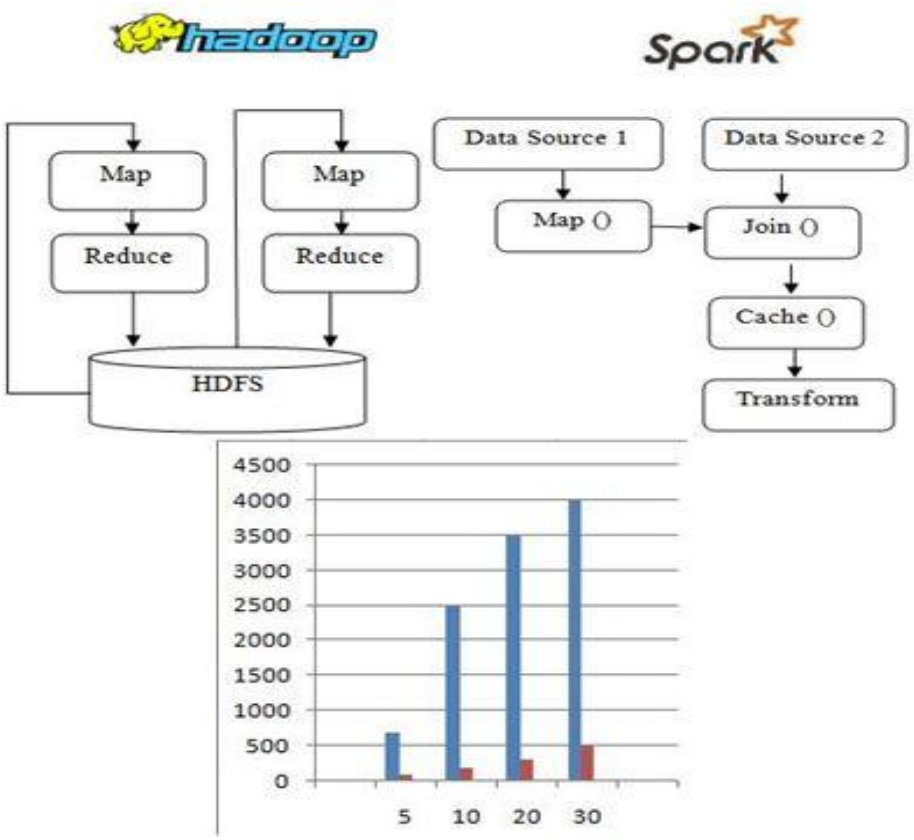

Fig.1 Contrast to Spark and MapReduce

1.2 ARCHITECTURE OF APACHE SPARK

There are three major standard factors in Sparkarchitecture. Data Storage, Programming Interface, and Resource Management.

The first factor of Apache Spark is data storage, it handles the HDFS structure for information accumulationoutcome. Apache Spark includes its work progress through Hadoop that allows works Cassandra, Hadoop Distributed File System [16], HBase, and so on.

January - February 2020
ISSN: 0193 - 4120 Page No. 1224 - 1231

The second factor of Apache Spark [20] is Programming Interface which provides spark based applications to design API applications and make use of its standard programming languages such as python, java, scala using API interface.

The third factor of Apache Spark is Resource management transfers message as a standalone server or in some prominent cases, it is used asMesos or YARN that follows a distributed computing framework [3].

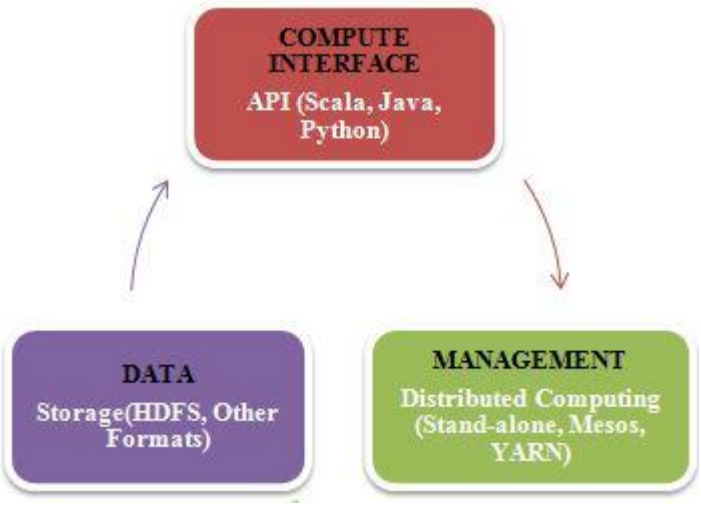

Fig.2:Components of spark architecture design.

1.3 SPARK SPECIFICATION

The data specified in Spark uses RDD (Resilient Distributed Dataset) which approves data computations on the cluster at distinct nodes. It helps to estimate the damaged data through fault tolerance and node failures. Data is distributed over manifold nodes. RDD parallelly[12] implements execution(faster than as usual MapReduce program i.e., 10,000 times). This process routinely preserves the data in memory and retainsthe existence of iterative algorithms related to machine learning [7].

The behavior of both Conventional MapReduce and Directed Acyclic Graphprocessing mechanism machinesis uncertain on particular applications that keep relying on the acyclic data stream, consist of stable storage, and also have evolutionaryanalyzing of data with the unmistakable task.

The effective speed in flash grantsus to function stream refinement with comprehensive info data and governmassive chunks of data on the glide. Similarly, this can also be promoted and takeadvantage of online machine learning. This process appoints theprerequisite use cases ofrepeatedscrutiny that develop to be anessentiallyubiquitous cause in the enterprise.

Inadequate use of multi-pass applications in MapReduceneeds low-latency data allocation over the diversified parallel process. These analytical applications are very primitive, and hold:

- Algorithms related to Iterative, encompassabundant machine learning methods and graph procedures like PageRank.
- This is also used to Iterative data mining,which consigns the client data into RAMin addition to the cluster and inspects it more than once.
- Streaming applications with a gradual change in time providesa quantity accumulated state.

2. IMPLEMENTATION

2.1 APPROACH TO K-MEANS CLUSTERING

A straightforward transparent K-Means clustering algorithm using clustering analysis. The primary intention is to select an elite partitioning of n entities in k cluster categories, on satisfying the distance

January - February 2020
ISSN: 0193 - 4120 Page No. 1224 - 1231

condition among the categorized members and its interrelated centroids, prototypical of the category, is lessened. Cluster center is estimated by all over all object's the mean value within the cluster. The methodology of the designed algorithm is as follows:

Step1: Create n number of clusters each and everyone objects that reside in clusters. Every cluster is labelled by allocating numbers.

Step2: Consider and estimate the distance among cluster objects defined as D(u,v). Where u,v were defined as the objects within the objects,(atu,v=1,2,...n.). Assume the square matrix for calculating distance as D= (D(u,v)). Incase vectors are shown on behalf of objects, Euclidean distance is applied.

Step3: Later, asset the largestcoincident match of clusters r and s, lest the distance, D(u,v), is lowestbetwixt all the duo wise interval.

Step4: Now fuse u and v to a unique cluster c and findthe distance among cluster distance D(c,k) for eachactual cluster k!=u,v.Obtained distances are observed andequivalent values from rows and columns are eliminated to the old cluster u and v in the defined matrix D, as u and v do not prevail further. Finally, concatenate new row and column in matrix D related to cluster c.

Step5: Periodically perform the process from step3 a result of n-1 times up to only single cluster is left

3. COMPARISON

The interrelationship betweenApache Spark and MapReducedesignated to reach a decision that has been carried out by testing andhandling all systems on a dataset that authorizesthe user to functionclusteringby applying the K-means estimation.

3.1 DATASET SPECIFICATION

This paper includesthe healthcare_sample_datasetsdataset with the size of 3.13 MB gatheredfrom recent years. Datasetstores Patient identification number (Patient_ID), Patient_Nameand Patient_DOBand other values information about particular records. Following table represents the data records and is testified in the table1:

Table 1: Patient records representation in Healthcare_sample_datasets

Patient_ID	int
Patient_Name	Chararray
Patient_DOB	Chararray
Patient_PhoneNumber	Chararray
Patient_emailAddress	Chararray
Patient_SSN	Chararray
Patient_Gender	Chararray
Patient_Disease	Chararray
Patient_weight	Float

Sample Record Values

211	Fa1	5478	Fa1@xx.com	11	M	Diabetes	72
212	Fa2	5478	Fa2@xx.com	11	F	PCOS	64

3.2 DATASET PERFORMANCE EVALUATION AND EXPLANATION

Samples from healthcare_sample_datasets dataset applythe K-means algorithm. This has resulted inthe following outputs described in the below table2 using comparison. To acquire a mixed evaluation, we examined 64MB and a single nodewith 3.13MB, two nodes with 3.13MB and supervise the efficiencybased on the conditions and its timingfor clustering as per the necessity employing K-Means procedure. Following are the specifications o the systems that perform Spark and MapReduce:

- The memory size of 4GB RAM
- The operating system as Linux Ubuntu
- Hard Drivewith 500 GB

Observing the obtained results from Apache spark has gained high speed in terms of time. It is identified that

January - February 2020
ISSN: 0193 - 4120 Page No. 1224 - 1231

depending on the dataset size Spark is 3times faster than MapReduce. Despite the minor inconstancy in this product,the K-means algorithm performs randomly and will not influence large quantities.

Table 2 Output for K-Means using Spark (MLib)

Size of the Dataset	Number of Nodes	Executed Time(s)
64MB	1	18
3.13 MB	1	149

Table 3 Output for K-Means using Map Reduce(Mahout)

Size of the Dataset Size	Number of Nodes	Executed Time (s)
64MB	1	44
3.13 MB	1	291
3.13 MB	2	163

For the assessment number of considered nodes and effective performance of spark and MapReduce, metrics like scheduling delay, speed up, energy consumption is measured for each cluster.

3.2.1 Analysis of Scheduling Delay using Spark vs.MapReduce

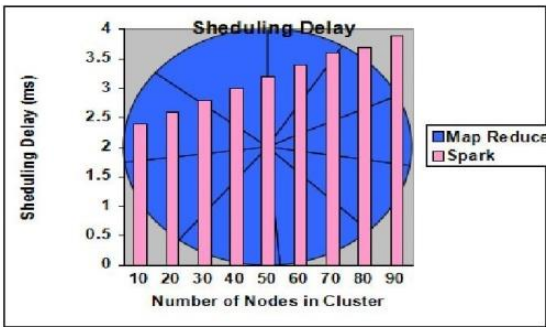

Fig.3: Cluster analysis over scheduling delay

Figure3 represents the Cluster analysis over scheduling delay with respect to the spark and map-reduce in the Hadoop. The spark is exhibitingabetter scheduling length in contrast withthe map reduce.

3.2.2 Analysis for the Speed up using Spark vs.MapReduce

The speedup was defined as the ratio of the sequential complete time of the schedule to the total length of the schedule obtained. Figure4 represents the Cluster analysis over speed up with respect to the spark and map-reduce approachesand its corresponding value continuously increases concern to the number of clusters.

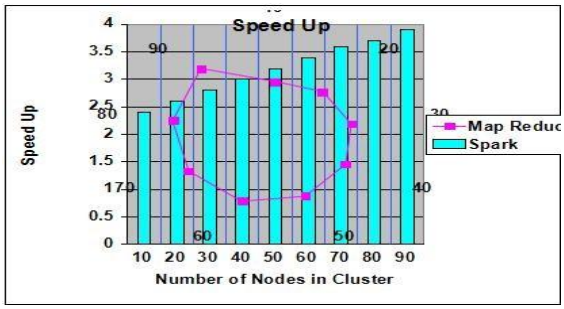

Fig.4: Cluster analysis over Speedup

3.2.2 Analysis for Energy Consumption using Spark vs.MapReduce

Figure5 represents the Cluster analysis over speed up with respect to the spark and map-reduce approach. It was identified that Spark has consumed less energy when compared to MapReduce. Cluster resource has continuously increased.

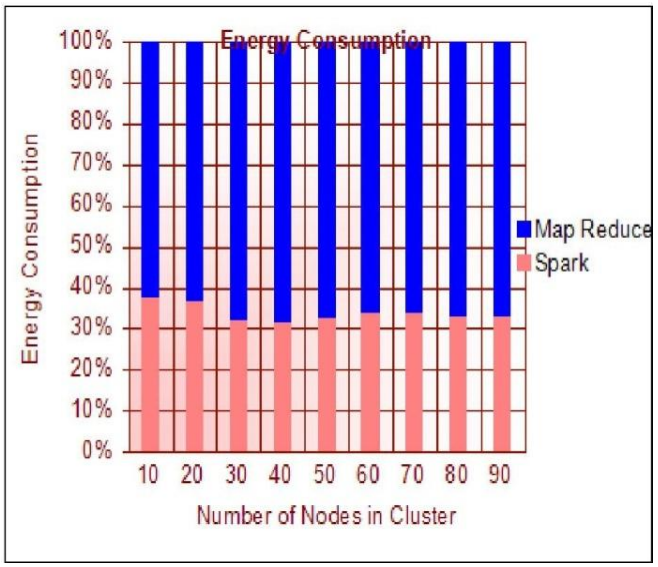

Fig.5: Cluster analysis over Energy consumption

January - February 2020
ISSN: 0193 - 4120 Page No. 1224 - 1231

Conclusion

This paper provides anaudit pair of the structuresthat moreover evaluates the different particular parameters that drawafterward an enforcement review and takes advantage of K-means estimation. The overall reaction for this exploration establishes that spark is remarkably solid foe and command beyond anambiguityto manage and modify through the resort as a part of flashbackpreparation. On observations carried out on spark capacity to functionon organized manipulations, spouting, and machine learning over unchanginggroup and business catching a quick look at the instant rate of receivingof spark. Spark has provided efficient solutions to countless cases among much other processing that includes Big Data preparing.

References

1. BayuPrabowoSutjiatmo, AfianErwinsyah, E. Laxmi Lydia, K. Shankar, PhongThanh Nguyen, wahidahHashim, AndinoMaseleno, "Empowering the Internet of Things (IoT) through Big Data", International Journal of Engineering and Advanced Technology (IJEAT), Vol.8, pg. 938-942, August 2019.
2. Muruganantham A., PhongThanh Nguyen, E. Laxmi Lydia, K. Shankar, WahidahHashim, AndinoMaseleno, "Big Data Analytics and intelligence: A perspective for Healthcare", International Journal of Engineering and Advanced Technology, Vol.8, pp.861-864, 2019.
3. Chen, Z., Xu, G., Mahalingam, V., Ge, L., Nguyen, J., Yu,W., Lu, C. "A cloud computing based network monitoring and threat detection system for critical infrastructures", Big Data Research, Vol.3, pp.10-23, 2016.
4. Celli, F., Cumbo, F., Weitschek, E. "Classification of large DNA Methylation datasets for identifying cancer drivers", Big Data Research, Vol.13, pp.21-28, 2018.
5. Subbu, K. P., Vasilakos, A,V. "Big Data for Context-Aware Computing - Perspectives and Challenges", Big Data Research, Vol.10, pp.33-43, 2017.
6. Milani, B, A., Navimipour, N. J. "A Systematic literature review of the data replication techniques in the cloud environments", Big Data Research, Vol.10, pp.1-7, 2017.
7. Elshawi, R., Sakr, S., Talia, D., Trunfio, P. "Big Data systems meet machine learning challenges: towards BigData science as a service", Big Data Research, Vol.14, pp.1-11, 2018.
8. Apiletti, D., Baralis, E., Cerquitelli, T., Garza, P., Pulvirenti, F., Michiardi, P. "A parallel MapReduce algorithm to efficiently support itemset mining on high dimensional data", Big Data Research, Vol.10,2017.
9. K Pavan Kumar, "An integrated health care system using IoT", International Journal Of Recent Technology and Engineering, Vol.7, ISSN: 2277-3878, 2019.
10. Dr. B. Premamayudu, LeelaPriya, "New reliability routing path for detects malicious", Ingeineri des susyem's d, Vol.24(2), 2019.
11. Apache Hive thhp://hadoop.apache.org/hiveScalaprogramming language. http://www.scala-lang.org.
12. C.Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins. Pig Latin: a not-so-foreign language for data
13. Y. Yu, M. Isard, D. Fetterly, M. Budiu, U. Erlingson, P. K. Gunda, and J. Currey. "DryadLINQ: A system for general-purpose distributed data-parallel computing using a high-level language" In OSDI '08, an Diego, CA, 2008.
14. J. Dean and S. Ghemawat. "MapReduce: Simplified data processing on large clusters". Commun. ACM, 51(1):107-113, 2008.
15. M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly. Dryad: Distributed data-parallel programs from sequential building blocks. In EuroSys 2007, pp.59-72, 2007.
16. B. Nitzberg and V. Lo, "Distributed shared memory: a survey of issues and algorithms", computer, 24(8):52-60, Aug 1991.
17. Elhoseny, M., Bian, G. B., Lakshmanaprabu, S. K., Shankar, K., Singh, A. K., & Wu, W. (2019). Effective features to classify ovarian cancer data in internet of medical things. Computer Networks, 159, 147-156.
18. Shankar, K., Elhoseny, M., Perumal, E., Ilayaraja, M., & Kumar, K. S. (2019). An Efficient Image Encryption Scheme Based on Signcryption Technique with Adaptive Elephant Herding Optimization. In Cybersecurity and Secure Information Systems (pp. 31-42). Springer, Cham.
19. Elhoseny, M., & Shankar, K. (2020). Energy efficient optimal routing for communication in VANETs via clustering model. In Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks (pp. 1-14). Springer, Cham.
20. Elhoseny, M., Shankar, K., & Uthayakumar, J. Intelligent Diagnostic Prediction and Classification System for Chronic Kidney Disease, Nature

January - February 2020
ISSN: 0193 - 4120 Page No. 1224 - 1231

Scientific Reports, July 2019. Press. DOI: https://doi. org/10.1038/s41598-019-46074-2.
21. Dutta, A. K., Elhoseny, M., Dahiya, V., & Shankar, K. (2019). An efficient hierarchical clustering protocol for multihop Internet of vehicles communication. Transactions on Emerging Telecommunications Technologies.
22. Elhoseny, M., & Shankar, K. (2019). Optimal bilateral filter and Convolutional Neural Network based denoising method of medical image measurements. Measurement, 143, 125-135.
23. Murugan, B. S., Elhoseny, M., Shankar, K., & Uthayakumar, J. (2019). Region-based scalable smart system for anomaly detection in pedestrian walkways. Computers & Electrical Engineering, 75, 146-160.
24. Famila, S., Jawahar, A., Sariga, A., & Shankar, K. (2019). Improved artificial bee colony optimization based clustering algorithm for SMART sensor environments. Peer-to-Peer Networking and Applications, 1-9.
25. Lakshmanaprabu, S. K., Shankar, K., Rani, S. S., Abdulhay, E., Arunkumar, N., Ramirez, G., & Uthayakumar, J. (2019). An effect of big data technology with ant colony optimization based routing in vehicular ad hoc networks: Towards smart cities. Journal of cleaner production, 217, 584-593.
26. Maheswari, P. U., Manickam, P., Kumar, K. S., Maseleno, A., & Shankar, K. Bat optimization algorithm with fuzzy based PIT sharing (BF-PIT) algorithm for Named Data Networking (NDN). Journal of Intelligent & Fuzzy Systems, (Preprint), 1-8.
27. Shankar, K., Ilayaraja, M., & Kumar, K. S. (2018). Technological Solutions for Health Care Protection and Services Through Internet Of Things (IoT). International Journal of Pure and Applied Mathematics, 118(7), 277-283.
28. Lakshmanaprabu, S. K., Shankar, K., Ilayaraja, M., Nasir, A. W., Vijayakumar, V., & Chilamkurti, N. (2019). Random forest for big data classification in the internet of things using optimal features. International Journal of Machine Learning and Cybernetics, 1-10.
29. Sankhwar, S., Gupta, D., Ramya, K. C., Rani, S. S., Shankar, K., & Lakshmanaprabu, S. K. (2016). Improved grey wolf optimization-based feature subset selection with fuzzy neural classifier for financial crisis prediction. Soft Computing, 1-10.
30. Iswanto, I., Lydia, E. L., Shankar, K., Nguyen, P. T., Hashim, W., & Maseleno, A. (2019). Identifying diseases and diagnosis using machine learning. International Journal of Engineering and Advanced Technology, 8(6 Special Issue 2), 978-981.
31. Lakshmanaprabu, S. K., Mohanty, S. N., Krishnamoorthy, S., Uthayakumar, J., & Shankar, K. (2019). Online clinical decision support system using optimal deep neural networks. Applied Soft Computing, 81, 105487.
32. Shankar, K., Lakshmanaprabu, S. K., Khanna, A., Tanwar, S., Rodrigues, J. J., & Roy, N. R. (2019). Alzheimer detection using Group Grey Wolf Optimization based features with convolutional classifier. Computers & Electrical Engineering, 77, 230-243.