International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-5, January 2020

Data Dissemination Techniques using DBSCAN and DD-Rtree for Spatial Data Mining

Basavaraj S. Prabha, Arun Biradar

***Abstract*: *In today's scenario where data volumes are growing on enormous speed over cloud or internet, we want to limit this growing data size. This can be achieved by data processing methods where data processing can be done in parallel. To make the data processing done in parallel, various clustering sampling methodologies are in use such as Slink, DBSCAN, and Optics and so on. The power accomplished by various methodologies which already exist will be focusing to the preservation of three-dimensional surroundings such as grid tree, grid files, quad tree and tree like k-d-tree, etc. This all compartmentalization constructions are generally done in static way which is a fix way. Since this data volume size is very big, this results in a high cost of information sharing and clustering. Hence through this research work we want to analyze various clustering algorithms both on static level and at dynamic level. For doing this we are majorly comparing the dynamic distribution using DBSCAN and DD-Rtree algorithm by proposing a DD-Rtree will help us to preserver the spatial vicinity. In addition, DD-Rtree is not static but more than that it is dynamic, i.e. it will create build the data as we progress with clustering. DD-Rtree methodologies are based on R-Tree concepts which analyses the data at dynamic random way. We tend to compare DD-RTree's information distribution norm with one of the clustering system recently published, DBSCAN. On the side of the potential of DBSCAN formula, we tend to distinguish the potential of queries managed by these compartmentalization structures. Numerous applications requires such kind of implementation at dynamic level of spatial database system such as satellite images, X-Ray crystallography, metrological department or other such atomic equipment's spatial datasets. Our research work will help to implements spatial data dynamically using DDR-tree mechanism.*

Keywords: Data Dissemination, KNN, Spatial Data Mining, Density Clustering

I. INTRODUCTION

Over the past few years, data is increasing at a rapid speed over the globe with increasing use and demand of the Internet leading to increase the need for Big Data analytics. To work better and efficient with Big Data across various data centers, mining the data becomes very essential. To achieve this, many research works had done and proposed on clustering the dataset on the big data platform. When we distribute the data using a data mining algorithm, the preservation of spatial locality is taken care of. Whenever we are trying to allocate and access the data spatially over different clusters we also need to take care of reducing the inter-node communication time. Some of the spatially manipulating algorithms with native knowledge are DBSCAN [1], OPTICS [2] [3], etc. Recently a lot of work was carried out to place these methodologies on top of native and entirely different clusters are being made [3] [4] [5] [6]. In terms of execution, these methodologies implements or follow a certain structure or working. The data provision stage performs a crucial role in improving the execution of an inversion steps. Many algorithms for data processing may need to perform neighborhood and neighborhood (K-NN) queries. Carrying out such questions for the purpose of knowledge p becomes cost-effective once regional access is made to the information needed by these queries. If this request is not met, we would like to obtain knowledge from alternative cipher nodes and thus acquire inter-node communication value.

Many diverse variants of decentralized information frameworks are projected in the research which includes — Parallel Rtree[7][8], Distributed B-link tree, Distributed Random tree, Master-Client R-tree, Upgraded Parallel R-tree[9], SD-RTREE[10], etc. Several of such knowledge systems aim to increase the level of communication in order to induce optimal results of questions. SD-RTREE [10] is the last distributed system proposed that reduces the large overhead of communication in the construction of the dataset and querying the big data set over the clusters. It internally uses the R-trees concept of implementation of minimum bounding rectangles (MBR) along with splits of R-tree node structures. It gives access dynamically and performs well in terms of scalability even if we add the cluster nodes. Nevertheless, since its redeployment policy relies on K-NN search, SD-RTREE [10] ensures adequate preservation of the 3-Dneighborhood. It also does not ensure a good balance of load. Therefore, an associated degree of efficient diverse distributed storage structure is needed that can satisfactorily maintain spatial section and ensure load equalization, thus helping in improving request efficiency in data processing procedures. DD-RTREE is a new active scattered arrangement constructed on R-tree, is the main interest of this article. It retains the section of space and ensures proper equalization of the load. Additionally, the design of DD-RTREE making it active meaning that the information is inserted gradually and that cipher nodes can be inserted progressively if required.

Manuscript published on January 30, 2020.
* Correspondence Author
Basavaraj S. Prabha*, Ph.D Research Scholar, CSE Department, CMR University, Bengaluru, India, bsprabha@yahoo.com
Dr. Arun Biradar, Professor & Head, Dept. of CSE, CMR University, Bengaluru, India, arunbiradar@hotmail.com

The construction of DDRTREE consists of 2-level R-trees. The first level of the R-tree is the index-R-tree (IR-TREE) and that is the index during the design / implantation process. The other level includes numerous R-trees for every cipher nodule (MRTREE), which buffers information about happiness to the node.

II. LITERATURE SURVEY

In various applications around the globe, in real time situation, one needs to manage spatial data. Increasingly large amount of data or information is derived from various satellite images, digital images from various devices, and hence automatic knowledge retrieval of information becomes important in spatial databases. Distributed data structure (DDS) is typically used in information sharing system in a cluster of computing nodes. For the task of identifying sophistication, grouping algorithms [11] are attractive. Application to big cognitive databases, however, makes clustering algorithms necessary since minimal domain knowledge criteria are required to define the information parameter, as correct values are still not recognized beforehand when looking at individual databases and an arbitrary shape cluster can be defined.

There can be two fundamental types of methodologies for clustering [12] as well as patriarchal algorithms. Reformatting Methods Identify a data constraint D of n items in a k sorting unit, k act as source parameter for these methodologies, i.e. it involves several subject matter expertise is required that is regrettably not accessible for many applications.

The clustering rule starts with the preliminary fragmentation of D and then uses the associated repeated management strategy to optimize the related goal value. Group is described by the force hub of the cluster, Or one of the subsystem components close to its origins (k-medoid frameworks). Partitioning algorithms then use a dancing technique for the ballroom. Next, confirm the success of the goal by the members of k. Collection is described through the gravitational middle of the collection or some of the entities of the collection close to the situation core (k-medoid frameworks) .The another phase means that a barrier is a kind of voronoi diagram and in all of the voronoi cells, each cluster is contained in one. Therefore, convexo-concave, which is very restrictive, is the form of all clusters identified by a partitioning law.

In their paper, Ng &han (1994) discusses KDD partitioning algorithms in relational datasets [13]. The related degree method known as CLARANS is implemented. CLARANS is more realistic and economical than previous k-medoid algorithms. Associate experimental analysis shows that CLARANS operates on millions of artifacts repositories expeditiously. Furthermore, researchers discuss tactics to see the "natural" cluster range knat in detail [13]. They are suggesting to run CLARANS between two ton once for every k. The silhouette factor is determined for each of the discovered clustering, andto end with the "natural" cluster with the most outline factor is preferred[12].

Unfortunately, the applied strategy of runtime is unaffordable for big size of n, as it involves CLARANS ' O(n) invites to reduce overhead communication, distributed data processing (DDM) approaches that comprise two main steps are projected. Because the knowledge is sometimes distributed the primary section consists of executing the mining method on native datasets on every node to make native results. These native results are aggregative to create international ones. So the efficiency of any DDM formula depends closely on the efficiency of its aggregation section. In this context, it became important to analyze these large and multi-dimensional datasets using distributed data processing (DDM) strategies with economic aggregation portion. In addition, DDM relates a lot to large-scale distributed networks, such as clusters and grids[14], whereby data sets are typically globally dispersed and closely managed by entirely new organizations.

Recent researches have planned distributed bunch approaches supported an equivalent 2-step process: Perform minimal native information analysis on third party sites and send them to a central website to build world models by aggregating native outcomes.

DDS consists of an information Management structure and a collection of decentralized control routes to allow computer entities to request queries and alteration orders and to receive appropriate responses. Its information structure serves as an overview for both the collections of data frameworks stored at either node [15]. Several DDSs, including 1-to-1 network layouts, data analytics, social media network mining, are proposed in literature etc. [15][16][17]. Usually these information systems are used to index information for proficient handling of requirements, course-plotting, etc. We do this by arranging data in such a way that many disk arrays can be reached at the same time to answer a query. They are hence not appropriate for the delivery of information. The new version is the SD-RTREE [10], a dual design based on the structure of AVL [18] and the structure of R. This facilitates complex insertions and demonstrates strong scalability. But it does not guarantee a good balance of load. The strategy used for re-distribution is often predicated on K-NN lookup that does not ensure that good areas of data were maintained. It is apparent from the review done in previous review, neither of the data types aim to preserve the spatial location in its distribution but at concurrently maintaining a strong workload distribution and providing optimal request efficiency for concurrent machine learning strategies.

They are used to distribute parallel algorithms across computer nodes in a cluster. A big downside of these datatypes was that they were constant, i.e. they essential to recited the complete collection before splitting; hence they are expensive to index big data. Patwary et al. [5] suggested sampling distribution methodologies to distribute large data. Nonetheless, this planning is highly costlier and has large overhead contact. There is also a static segregating system here.

A method is proposed to maintain a spatial location based on Vornoi diagrams, in addition to tree-based data structures [19]. This scheme, however, is also stationary and must examination the whole dataset is done before it is built.

III. DBSCAN ALGORITHM

DBSCAN (Density-based abstraction cluster of Application) could be higher performance information cluster algorithmic rule. It offers the next performance rate for information wherever a continuing density of information points is allotted in clusters. Noise reduction is one of the various attributes of this algorithmic law. Clustering which depends on cluster is that the approach of characteristic distinctive teams or clusters during giant information set was based on the notion that a cluster is also a dense contiguous region within the complete data house, separated by relatively low information density adjacent areas from specific clusters. Knowledge points with a relatively lower number of artifacts within the splitting regions are commonly known as noise or outliers. Out of the many cluster algorithms, DBSCAN is one among the foremost used and powerful cluster algorithmic rule supported density-based cluster that is employed to seek out the accuracy of cluster that varied in form and size in huge datasets which could have some noise. Additionally one among the key points that create DBSCAN distinctive is that it does not would like early identification of many clusters on oversized information.

Two necessary things are the basic needs of implanting the DBSCAN algorithmic rule

1) EPS, it is the neighborhood's radius of sampled knowledge points.

2) Min Pts is the minimum adjacent range of points of information within the region.

Although users do not need ample pre-knowledge of the range of groups in the datasets, users decide Eps and MinPts. Since the dataset differs in a very giant range from each other, it is difficult to manually predict an effective Eps value. Nevertheless, since Eps plays a major role in the outcomes of the DBSCAN algorithmic law, the results for different Eps values may vary drastically. As a result, DBSCAN's overall performance degrades significantly when clusters with different densities of information are identified as results of the Eps values.

Since we prefer to use foreign amounts for mainly Eps and also MinPts, we may combine in DBSCAN may 2 groups into only single group if two groups of diverse concentrations are "similar" to any alternative. Let's the gap among 2 ranges of objects o1 and o2 be described as dist(o1,o2)=min If the difference between the 2 sets is greater than Eps, 2 collections of arguments getting at tiniest size of the smallest collection will be detached from each another. Therefore, DBSCAN's algorithmic decision is also appropriate for the clusters identified with a better price for MinPts. Nevertheless, due to the algorithmic implementation of DBSCAN, this does not result in a trendy and extremely economical basic rule.

In DBSCAN, let subset be referred to as D, the clustering radius of the algorithm, Eps, so the required number of items in the area of Eps, MinPts, and the following definitions describe the basic concepts of the algorithmic law:Eps-neighborhood: The Eps-neighborhood of that time is the neighboring data point within the Eps radius of a given point within the dataset.. Core purpose: If a point view of the Eps neighborhood contains a lowest possible range of neighbor's, MinPts, then a core point is taken into consideration in the intent of the test.

Border point: If the Eps-neighborhood of a check point does not contain a sufficient variety of neighbors, MinPts, however, is considered to share its Eps-neighborhood with at least one known core purpose, and then the object of the test is called a border purpose. Directly attainable density: p is deemed to be directly attainable by density from another alphabetic character meaning if p is recognized as a core point among Eps ' alphabetic type neighborhood and alphanumeric character.

Figure 1 demonstrates the concept of direct density reachability

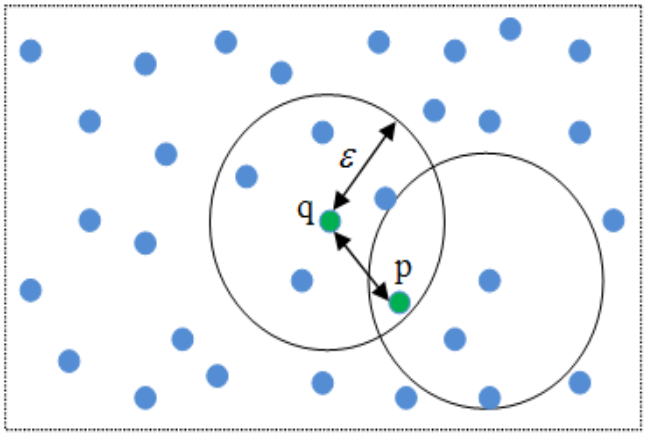

Figure 1: Direct Density Reachability in DBSCAN

Working of DBSCAN is as follows

• DBSCAN start with a selection of a random point

In the density-reachable possible points from p with a distance, Eps(ε) known as MinPts, selection is accordingly done.

• P is considered as main or basic point, and then we forms the clusters analyzed as Eps and MinPts.

• Incase p is considered a boundary juncture in that no density-reachable points are found from p and DBSCAN moves to the database's subsequent data.

Algorithm of DBSCAN

1. Start
 // define d as end point of every cluster
2. Set D= {End point of each segment}
3. Track all of the core points in D where
// check whether EPS>MinPts, if yes add CP in clusters
4. EPS>MinPts
5. Join core points with clusters
 // identify every BP where EPS<MinPts
6. Detect all border points which have
7. EPS<MinPts
8. Neighbors of basic points
 // Add BP to KP
5. Add boundary points to key points
6. Label other points as noise in D
7. END

IV. SD-RTREE

SD-Rtree is commonly used in a dispersedrecords system that operates on the idea of a cluster of data nodes of computing. It's like the AVL vine. The architecture implementation of SD-Rtree is illustrated in Figure 2. It is a binary tree length-balanced map to n server set that meets the below requirements.

• Each inner node has essentially two children (called the routing node). - Internal node has rectangles (dr) in the left and right folders, referring to the MBRs of the left and right subtrees.

.• Every foliagenodule (known as the data node) supplies the data point indexes stored on computer.

SD-RTREE has n joins and n-1 inner nodes distributed between n computers. Every mechanism in the cluster holds a pair (rni, dni) of rni as a direction-finding node and a dnirecords node. The height of each rni is stored; dr; two references directing to his leftward and rightward children; his parental id; and the intersecting coverage(OC). OC is a related level array containing its dr elements that intersect with appropriate machines. The height of each rni is stored; dr; two references pointing to his left and right children; his parent id; and the intersecting coverage (OC). OC is a related level array containing its dr elements that intersect with appropriate machines. For a native R-tree, the records deposited at the records nodes are indexed. That data node has the maximum capacity that it can index, maximum number of data points.

In addition, SD-RTREE preserves a picture I of the distribution tree accessed by the applicant user before the relevant insertion or query is triggered.

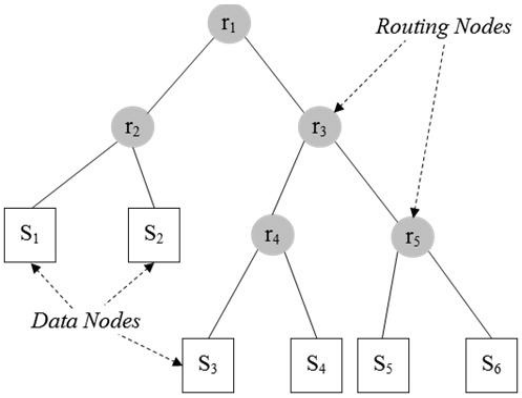

Figure 2: SD-RTree Structure

The picture supports to guide the associate entry or query message within the cluster to the correct computer. The image normally exists in a site in which an request will call all its activities. It is retained / upgraded from devices that are affected by any image adjustment messages implanting or deleting data sets (IAM).

V. DD-RTREE

DD-RTREE can be a diverse decentralized arrangement in a computer node cluster. The aim of DD-RTREE is to disseminate information across numerous reason points with the following goals: optimizing spatial location; achieving smart workload equilibrium; minimizing internode contact for its design and spatial and query period minimization analysis processing strategies. DD-RTREE is initially a three-dimensional compartmentalization system that aims to achieve the goals. Additionally, DD-RTREE's planning styles it diverse, i.e. the records will be staggering supplementary, also reason access points can be incrementally supplementary if necessary.

A. Model of DD-RTREE

DD-RTREE contains all types of R-trees. The main level of the R-tree is index-R-tree (IR-TREE), which is the Key of the overall structure; exist in in a master purpose nodule or repository wherever the whole area group of directions is given. Another level contains multiple R-trees at every cluster system (MR-TREE) holding on each. MRTREE indexes its machine's data points. IR-TREE meets the subsequent things:

• That IR-TREE node has a least number of I m and a extreme number of IM entrances keys on it, apart from the core that may be smaller thanI m entries..

• Every inner nodule involves of MBRs that hoard the boundary info of entirely indexed entities on their particular sub trees.

• Every extrinsic nodule shops all notes with MBR information keys in a machine. That is to say, it stores an MR-TREE's root MBR. It also supplies the mechanism ID of the machine where the MR-TREE is deposited and an indexed number of ideas (cnt).

• The outer node also comprises a stable size buffer that preserves data points temporarily before moving them hooked on the exact MR-TREE.

MR-TREES are R-trees having properties of the native R-tree. Every device has a mc capability that is the largest quantity of datasets that this may be keyed.

DD-RTREE advantages: The DD-RTREE layout accomplishes limited overlap between the machine's bounding rectangles. That's because; all the methods governing DDRTREE development are focused on the concepts of R-tree design. DDRTREE displays good location space and efficient performance of queries. Buffers attached to the IR-TREE leaves make it possible to reduce overhead communication during the construction phase. While DD-RTREE's restructuring strategy requires high overhead communication;

it is expected to be more effective because the overall number of reallocation compared to DBSCAN is much smaller. As a matter of fact, the design time at DD-RTREE is less. Furthermore, the reallocation policy is created on the ideologies of reducing the overlap between the Boundaries Square of the technologies particularly in comparison with DBSCAN based on K-NN. It therefore provides better delivery position and efficient request efficiency.

DD-RTREE acts as an efficient way of data transmission through computer nodes within a group to increase the efficiency of concurrent spatial analysis processing techniques.

VI. METHODOLOGIES

A: Data Distribution Approach:

In step 1, we used an appropriate method to distribute data. DD-RTREE takes less time in this process when compared to DBSCAN.

B: Clustering of Data

Phase 2, For each computer, one can collect information ideas from extra devices in the region located within the current computer and the extended boundary. In this phase, the number of necessary MPI communications in all situations residues almost similar. The running period for DD-RTREE is lower, however, since if the quantity of additional data points from other devices are smaller. Point here to note that the number of points obtained from other devices is very low in tree-based propagation which has DD-RTREE. This is because the machines are completely disconnected in their distribution.

C: Implementing KNN approach on clustered Data:

Step 3 includes the execution on each computer of local DBSCAN. In this stage, the interval needed for native DBSCAN is smaller for DD-RTREE and k-d-tree-based dissemination, as the smaller number of additional points on each point helps to reduce the time required for neighborhood queries.

D: Combination of data over clusters

In step 4, we combine all the localized DBSCAN results to provide the requisite universal clustering.

Algorithm:D-TREE-PARTITIONING

Procedure is defined as:
1. Distribute data with all data points randomly in the clusters of machines
2: Choose a small set of pieces of data of each machine then send data to certain machine clusters.
3. From the sample of every data collected, calculate median
4: Clustering or partitioning of data is done in two parts , one on leftward side of median and other on right way side of median
5: Exchange data between two machines so that another device is left-handed and another machine is left-handed.
6: Repeat step 2-5 on all machines of left side and also for right side as well recursively.
7: SO through this algorithm partitioning of data over clusters using DDR-Tree is implemented

VII. PERFORMANCE EVALUATION

Boundaries Square of the technologies particularly in comparison with DBSCAN based on K-NN. It therefore provides better to compare their efficiency across MPAGD16M3D datasets over 32 computers, we run a basic DBSCAN and DDRTREE version. We had evaluated the performance in terms of the timing execution required for clustering for both algorithms and their accuracy. Selected as 0.01 and Min Pts was selected as 5.
A description of its execution is presented in Table 1. DBSCAN detects all groups and the databases interference points. The time needed for combining in this phase is approximately same can be said for DD-RTREE and k-d-tree spreads use DD-RTREE and enhanced than DBSCAN. This demonstrations means that efficient data Distribution helps to reduce coordination complexity and to enhance parallel execution of the DBSCAN algorithms.

Table 1: Time Comparison between DBSCAN and DDR-Tree

	DBSCAN(Sec)	DD-Rtree(Sec)
Step 1	**1987**	**1345**
Step 2	**480**	**298**
Step 3	**1560**	**1398**
Step 4	**150**	**139**
Total Execution Time	**4177**	**3180**

VIII. CONCLUSION AND FUTURE WORK

Our research methodology in this research work compares the two spatial database clustering algorithm namely DBSCAN and DD-Rtree. Our research work suggests DD-RTREE, based on R-tree supported by a dynamic decentralized organization. DD-RTREE maintains its allocation of the spatial neighborhood, Realizes fair load reconciliation, shows less overhead communication in inquiring and building, and enhances the efficiency of algorithmic similar three-dimensional data storage rules after comparing the results with the DBSCAN spatial data storage method, then the overall performance time in DD-Rtree execution is considerably smaller. For mining information sources, DD-RTREE will be accustomed to designing an extremely economical distributed system.
The DDRTREE technique will also refer to different R-tree versions. DD-RTREE also facilitates the successful execution of queries concerning Ɛ-neighbourhood and KNN. It also allows for efficient distribution of k-d-tree over cluster nodes for very large datasets. Together, the consistency and efficiency assessment confirms DDRTREE's dominance over DBSCAN and random distribution. DD-RTREE can be used to develop highly effective distributed systems for data streams in the mining sector.

REFERENCES

1. M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A Density-BasedAlgorithm for Discovering Clusters in Large Spatial Databases withNoise," Second International Conference on Knowledge Discoveryand Data Mining, pp. 226–231, 1996.
2. M. Ankerst, M. M. Breunig, H.-P.Kriegel, and J. Sander, "OPTICS:Ordering Points To Identify the Clustering Structure," ACM SIGMODRecord, vol. 28, no. 2, pp. 49–60, Jun. 1999.

3. P. Goyal, S. Kumari, D. Kumar, S. Balasubramaniam, N. Goyal, S. Islam, and J. S. Challa, "Parallelizing OPTICS for Commodity Clusters," in Proceedings of the 2015 International Conference on Distributed Computing and Networking - ICDCN '15, 2015, pp. 1–10
4. W. Hendrix, M. M. Ali Patwary, A. Agrawal, W. K. Liao, and A. Choudhary, "Parallel hierarchical clustering on shared memory platforms," in 2012 19th International Conference on High Performance Computing, HiPC 2012, 2012, pp. 1–9
5. M. M. A. Patwary, D. Palsetia, A. Agrawal, W. K. Liao, F. Manne, and A. Choudhary, "A new scalable parallel DBSCAN algorithm using the disjoint-set data structure," in International Conference for High Performance Computing, Networking, Storage and Analysis, SC, 2012, pp. 1–11.
6. M. M. A. Patwary, S. Byna, N. R. Satish, N. Sundaram, Z. Lukic, V. Roytershteyn, M. J. Anderson, Y. Yao, Prabhat, and P. Dubey, "BDCATS: big data clustering at trillion particle scale.," in SC15: International Conference for High Performance Computing, Networking, Storage and Analysis'15, 2015, pp. 1–12.
7. I.Kamel,C.Faloutsos, "Parallel R-Trees" in SIGMOD '92 Proceeding of 1992 ACM SIGMOD international conference on Management of data, 1992, pp 195-204.
8. E.G. Hoel and H.Samet, "Data-Parallel R-Tree Algorithms ", in 1993 International conference on Parallel Processing-ICPP '93,1993,00 49-52.
9. S.Lai, F.Zhu,Y.Sun "A Design of Parallel R-Tree on cluster of workstations", in International Workshop Databases in Networked Information System,2000, pp 119-133.
10. C.DuMouza,W.Litwin and P.Riguax, "Large scale indexing of spatila data in distributed repositories: The SD-Rtree." VLDB Journal, vol 18, no 4,pp. 933-958, 2009.
11. Matheus C.J.; Chan P.K.; and Paitetsky-Shapiro G. 1993. System for knowledge Discovery in databases, IEEE transactions on knowledge and data engineering 5(6): pp 903-913
12. Kaufman L. and Rouseeuw P.J. 1990. Finding Groups in Data: An Introduction to cluster analysis.John Wiley and Sons.
13. Ng R.T. and Han J. 1994. Efficient and Effective clustering methods for spatial data mining, Proc. 20th Int. Conf. On Very Large Databases, pp 144-155, Santiago,Chile.
14. I. Foster and C. Kesselman, The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1999
15. E.N.Adriano Di Pasquale, "Scalable Distributed Data Structure: A Survey", 3rd International Workshop on Distributed Data and structures(WDAS '00), pp 9-16
16. K Brigitte and W.Peter "Distributing a search tree among a growing number of processors", in SIGMOD '94 Proceeding of the 1994 ACM SIGMOD international conference on Management of data,1994, pp 265-276.
17. M.T. Goodrich, M.J.Nelson and J.Z.Sun "The rainbow skip graph:A Fault-tolerant constant-degree distributed data structure" in proceeding of 17th annual ACM-SIAM symposium on discrete algorithm-SODA 06,2006, pp-384-393.
18. T.H. Cormen,C.Stein. R.L.Rivest,C.E. Leiserson ,"Introduction to algorithms ", Jul 2001.
19. A. Akdogan, U.Demiryurek, F. Baranei-Kashani and C. Shahabi "Vornoi-Based Geospatial Query processing with map reduce", in 2010 IEEE Second International Conference on Cloud Commuting Technology and Science, 2010,pp 9-16.

AUTHORS PROFILE

Basavaraj S. Prabha is currently a Research scholar at CMR University bengaluru having more than 12+ years of experience in teaching and 10 years in IT industry He worked on contact centre domain, network security and BIOS to name few areas. He completed B.E. in Computer science and Engineering from Gulbarga University, gulbarga and M.Tech in computer science and Engineering from Visvesvaraya Technological University Belgaum. His research interests are in the fields of networking, machine learning and data mining. He is a member of ISTE and IETE.

Dr. Arun Biradar is an academician having more than 22 years of experience in teaching, development, administration and leadership. He completed B.E. in Computer science and Engineering from Gulbarga University and M.Tech in Software Engineering from Mysore University and Ph.D. in Computer Science Engineering from Swami Ramanand Teerth Marathwada University, Nanded. His research interests are Wireless Ad-hoc Networks, Computer Networks, Software Engineering, Genetic Algorithms, Machine Learning, IoT and Cloud Computing. He is a recognised guide under Visvesvaraya Technological University, Belagavi, and has many students pursuing their Ph.D. under his guidance. He is a member of various professional bodies and has been a part of the NAAC and NBA Advisory Committee to many Colleges. He is an active member of IEEE.