Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 5, May 2014, pg.1013 – 1021

RESEARCH ARTICLE

An Efficient Algorithm for Clustering Data Using Map-Reduce Approach

Puppala Priyanka[1], SK.Abdul Nabi[2], Meena Kumari P[3]

[1]Department of CSE, AVN Inst. of Engg. & Tech., Hyderabad

[2]Professor & HOD, Department of CSE, AVN Inst. Of Engg. & Tech., Hyderabad

[3]Department of CSE, AVN Inst. Of Engg. &Tech., Hyderabad

[1] priya.puppala9@gmail.com

[2] nabi.cse@gmail.com

[3] meenasri33@gmail.com

Abstract— We have been studying the problem of clustering data objects. As we have implemented a new algorithm EMaRC which is An Efficient Map Reduce algorithm for Clustering Data. In clusters Feature selection is the most important part of the clustering process that involves and identifying the set of features of a subset, at which they produces accurate and accordant results with the original set of features. The main concept behind this paper is that, to give the effective outcomes of clustering features. In this the nature of clustering and some more concepts serves for processing large data sets. A map-reduce concept is involved followed by feature selection algorithm which affects the entire process of clustering to get the most effective and features produces efficiently. While efficiency concerns, the time complexity is desirable component, which the time required to find effective features, where effectiveness is related to the quality of the features of subsets. Based on these criteria, a cluster based map-reduce feature selection approach, is proposed and evaluated in this paper.

Keywords— Feature subset selection, filter method, feature clustering, map-reduce, EMaRC Algorithm

I. INTRODUCTION

Generally, data mining is defined as the process of analyzing data from different panorama and epitomizing it into useful information and sometimes called data or knowledge discovery Often the value of data mining applications is estimated to be immense. Most of the organizations have stored huge amounts of data over long periods of operation; also data mining is able to extract very valuable knowledge from this data. A cluster is a subset of data which are similar. Clustering, also known as unsupervised learning, which is defined as -is the process of dividing datasets into smaller groups such that the members of each small group are as similar as well as close as possible to one another. Different groups are as dissimilar (far) as possible from one another. Commonly used functional programming is inspired by the map-reduce functions and consists of two distinctive stages. In the beginning stage the selected clustering algorithm is applied (mapped) to each experiment, separately. In contrast to many conventional clusters based algorithms, to produce clustering feature outputs, there only one single dataset is used. Whereas we introduce a Map Reduce approach for clustering, which were generated in multiple experimental

settings. Mostly in business, cluster analysis can be used to discover and characterize customer segments for marketing purposes and in the biology; it can also be used for the classification of animals and plants given their features.

Two important types of clustering algorithms are:

1. Hierarchical
   a. Agglomerative
   b. Divisive
2. Partitioned
   a. K Means
   b. Self-organizing Map

Requirements for a good clustering method are:

- Within the cluster is similar
- Ability to deal with various kinds of attributes.
- Also deal with noise data and outliers.
- Between the cluster dissimilarity.
- It can handle high dimensionality kinds.
- And Scalability, Usability and Interpretable.
- Have the ability to find some or all of the hidden clusters.

The most important issue in the clustering is that - how to determine the similarity between two objects, so that within clusters, they can be formed from objects with high similarity and low similarity between clusters. Generally, to measure similarity or dissimilarity between objects, a distance measure such as Manhattan, Minkowski and Euclidean are used. The distance function returns a lower bounded value for a pair of objects which are much similar to one another. Partitioning of sets of data or a data set into similar subsets is known to be Data Clustering. During the process of data clustering a method is often required to ascertain how a group of objects or data sets or one object is similar to another. The data clustering method is generally comprehended by some kind of distance measure. A common technique is used in data analysis and also utilized in many disciplines which include data mining, image analysis and statistics that it defined as Data Clustering.

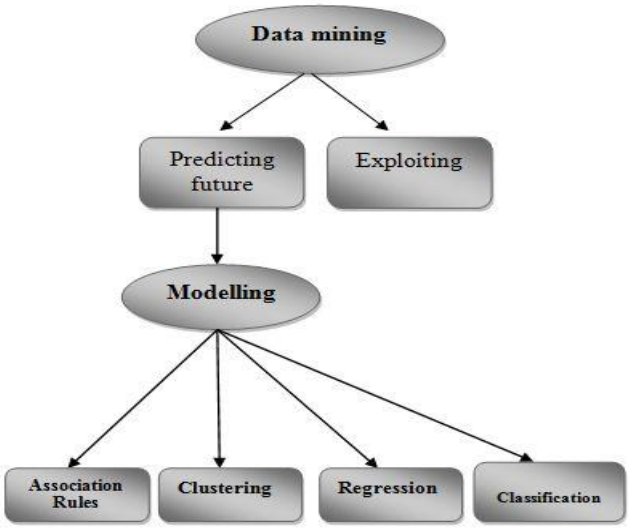

Fig.1.What is Data Mining& Classification

There are some kinds of clustering algorithms. The concept behind Hierarchical algorithms is that build consecutive clusters by using formerly defined clusters. Hierarchical algorithms can be agglomerative in nature that is accompanied by a bottom-up approach, which means they build clusters by consecutively or successively by merging the smaller ones. Besides, it can be divisive nature, is followed by top-down approach, which meant consecutively by splitting large data clusters into smaller groups. In terms of space and time complexity, data

clustering can be computationally more expensive. In addition to this, by repeating the data clustering further expense might be acquired. Therefore, in terms of the increased quantity of memory which is available in computing clusters and also computation speedup of distributing and parallelizing data clustering responsibilities becomes more attractive.

A framework is introduced, for which is useful in resolving various kinds of distributed problems by using computing cluster is Map-Reduce. Map-Reduce framework is designed with the most elementary kinds of mapping and reduces functionalities, that which is made in two steps. In Map-Reduce framework, that it is made by two simplest step process. Which are mapped and reduce steps, in first most steps in the framework a node can divide a pattern into several numbers of independent and identical parts which are specified to map tasks. Each and every map task can process its pattern and the resultant outputs in the form of key-value pairs. After the, map task in the second step of Map-Reduce the reduce part can have the result pairs from map task and then the particular pair-key will be processed. The power of the framework comes from the fact, that Map - Reduce tasks can be spread across the different nodes of the pattern. Thus, by designing the Map-Reduce framework, it is alleged to be Distributed Sorting Platform.

II. DEFINITIONS

The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. The task of discovering structures and groups are in the same way or in the data that are "similar", without using experienced structures in the data is known as Clustering.

Hierarchical algorithms find successive clusters using previously determined clusters. Hierarchical algorithms can be either agglomerative (bottom-up) or divisive (top down). Agglomerative algorithms begin with each object as singleton clusters and successively merge those with other clusters to create the final clusters. Divisive algorithms start with the entire data set of objects and partition the set successively in smaller clusters. Map Reduce is a framework that allows certain kinds of problems, particularly those involving large data sets to be computed using many computers.

a. Pattern Representation

Pattern representation refers to the number of classes, the number of available patterns, and the number, type, and scurf of the features available to the clustering algorithm. Some of this data may not be controllable by the practitioner.

The characteristics can be subdivided into the following-

- Quantitative features: e.g.
- Continuous values (e.g., weight);
- Discrete values (e.g., the number of computers);
- Interval values (e.g., the duration of an event).

The pattern represents knowledge if it is well read by humans, valid on test information with some grade of certainty, potentially useful, fresh, or validates a hunch about which the user was curious. Measures of pattern interestingness, either objective or subjective, can be applied to conduct the discovery procedure. Data mining systems can be separated according to the varieties of databases mined, the kinds of knowledge mined, or the techniques employed. Efficient and effective data mining in large databases poses numerous requirements and great challenges to researchers and developers. The issues involved include data mining methodology, user-interaction, performance and scalability, and the processing of a large assortment of informative characters.

Are all of the patterns interesting?

A data mining system has the potential to get thousands or even one thousand thousand of patterns, or rules. Are all of the patterns interesting? Typically not | only a minor fraction of the patterns potentially generated would actually be of interest to any given user.

Performance Issues

These include parallelization, efficiency and scalability data mining algorithms.

i. Scalability and Efficiency of data mining Algorithms

To effectively extract information from a vast amount of data in databases, data mining algorithms must be client and scalability. That is, the running time of a data mining algorithm must be predictable and acceptable in large databases. Algorithms with exponential or even medium-order polynomial complexity will not be of practical

function

i. Parallel, distributed, and incremental updating algorithms

The huge size of many databases, the extensive dispersion of data, and the computational complexity of some data mining methods are factors motivating the development of parallel and distributed data mining algorithms. Such algorithms cluster the data into partitions, which are processed in parallel. The answers from the partitions are then combined.

III. FEATURE SELECTION/ FEATURE EXTRACTION

Feature selection is the process of identifying the most effective subset of the original features to use in Clustering. Feature extraction is the function of single or more transformations of the input features to produce new most important features. Moreover or both of these techniques can be applied to hold an appropriate set of features to use in bundling. Feature selection, also recognized as variable collection, featured variable subset selection, is the process of pulling out a subset of relevant characteristics for function in model building. The central assumption when using a feature selection technique is that the data contain many redundant or irrelevant features. Special features are those which provide no more information than the currently selected features, and an irrelevant feature doesn't provide useful information in any framework. Feature selection techniques are defined as a subset of the more general field of feature extraction. Feature selection can be divide into four categories; Wrapper, Filter, Hybrid and Embedded methods. Where wrapper method uses determined knowledge algorithm to evaluate selected feature subsets that are optimized for the learning process.

IV. MEASURE OF SIMILARITY AND DISSIMILARITY

Clustering algorithms rely on the ability to find out whether two objects are similar to each other. For this, they use the notation of a distance function, where this function takes two objects as input and returns a positive real number corresponding to the distance between the two objects- the smaller the distance, the more alike the two objects are to each other. Several popular functions are available and one of these needed to be chosen for each clustering problem depending on the data types in that specific application. The length function is chosen based on the type of attributes used to identify an instance in a clustering problem.

V. MAP REDUCE STRATEGY

Complex problems such as the one being taken in this report must often be done in multiple Map Reduce steps. Each step takes as input the output from a previous step MapReduce. Data Preparation: This data set is a collection of huge amount of files each containing data for a single record. Each of these files contains the record identification number of the record as the first line.

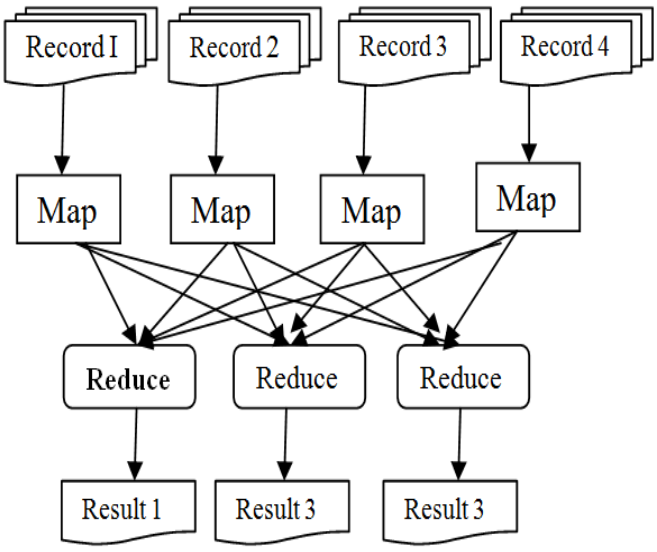

Fig.2.Map-Reduce Functionality

Map step:

The average output of the map will be recorded ID as the key and retired as the value. Every mapper maintains a collection bearing the canopy center candidates it has learned thus far. During every map the mapper determines if each successive record is within the distance threshold of any already determined canopy center candidate. The intermediate output sent to the reducer has the record ID as the cay and the list of retired-rating pairs as the value.

***Reduce Step*:**

The yield of the reduce step will simply output record ID as the key and concatenate the rater IDs for that record into a comma separated list. The reducer repeats the same procedure as the mappers. It meets the candidate canopy center record IDs, but takes out those which are inside the same threshold limit . In other words, it removes duplicate candidates for the same canopy center. In order for this to operate correctly the number of reducers is set to one.

a. Algorithm Implementation

As we have implemented a new algorithm EMaRC, which is An Efficient Map Reduce Algorithm for Clustering data. A new module is introduced in the first step is a Map, which can split the existing data clusters and another module is reduce can merge the intermediate data which is successful results of map phase or module. Then the intermediate data can be processed in reduce phase can give the efficient features. When map function is predicted, it processed the clustered data with key and value pairs and when the reduction is addressed then the intermediate dat a can be processed with grouped key-value collection, and merger function can be performed.

Algorithm 1: EMaRC algorithm with Map-Reduce

MaP FUNCTION {(Key,Value)/*(Record_id ,Record_value), T_β*/ }
Input: A cluster data (Key,value)
Output: An Intermediate Data sets $T_d \forall d$
1: MaP((const Key &key)T_β),/* (Record$_{id}$Record_value) */
{ //mapping (split) key pairs
2: FOR each { key, value, $\widehat{key}$ } in {(Key, Value), T_d }
3: P_d=f(key, T_β)
4: FOR each key $\in \widehat{key}$
5: Emit$(\widehat{key}, P_d)$ in the Intermediate Data
6: $T_d = T_d \cup (key, value, P_d)$
7: }
REDUCE FUNCTION {(Key, $\widehat{\boldsymbol{key}}$,Value)/*(Record_id ,Record_value), T_β*/ }
Input: An Intermediate Data sets T_d
Output: Estimated Coefficient E_β
1: $\boldsymbol{REDUCE}\ \{(\widehat{\boldsymbol{key}}, \mathbf{value}), P_d)\}$ //Intermediate Data
2: {
3: $E_\beta = \{(\widehat{\boldsymbol{key}}, \mathbf{value}), P_d\}$
4: *FOR* each $\{ (\widehat{\boldsymbol{key}}, \mathbf{value}) \in E_\beta \}$
5: $E_\beta = T_d \cap (\widehat{\boldsymbol{key}}, P_d)$ //Estimated Coefficient
6: }

Algorithm 1: EMaRC Algorithm

Where $(\mathbf{key}, \mathbf{value})$are the initial cluster data pairs

T_d is the total data sets as input

P_d is the Intermediate data $(\widehat{\boldsymbol{key}}, \mathbf{value})$

E_β is Estimated Coefficient (Efficient outputs)

$\widehat{\boldsymbol{key}}$ is key pair of intermediate data to be reduce

How many maps and reduces?

An interesting subject when using Map Reduce frameworks is determining how many maps and scales down to use for optimal operation. This is useful if one requires recognizing the number of output files for to be used as input in another MapReduce, but also can be useful if specifying the number of contracts is essential for ensuring correct results for some calculation. An instance of this is the Canopy Selection Map Reduce where the number of reduces had to be set to one in order for it to function right. Any machine who does not respond is taken "dead". Both Map- and Reduce-Machines Any task in progress gets needs to be re-executed and becomes eligible for scheduling. Map-Machines completed tasks are also reset because the answers are stacked away on local disk. Reduce-Machines notified to bring data from new machine assigned to take on the job. Problems suitable for processing with Map Reduce must usually be easily separate into independent subtasks that can be treated in parallel. In parallel computing, such problems as known as embarrassingly parallel and are ideally fitted to broadcast programming.

The power of Map Reduce is from the execution of many map tasks which work in parallel on a data set and these outputs the processed data in intermediate key-value pairs. Every reduce phasse only receives and processes data for one particular key at a time and outputs the information it processes as key-value pairs. Hence, Map Reduce in its most basic usage is a distributed sorting framework. Map Reduce is attractive because it permits a coder to write software for implementation on a computing cluster with small knowledge of parallel or distributed computing. Using more reduce tasks lower the cost of failures, but increases overhead of the framework and load balancing.

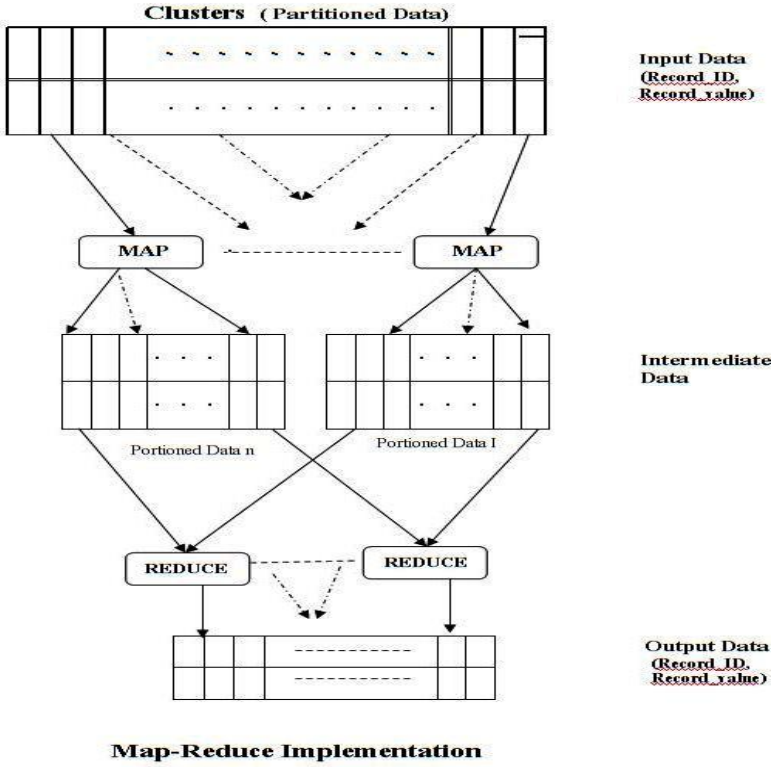

Fig.3. Map-Reduce Implementation

a) Joins:

Join is a popular operator that is not so well dealt with by Map and Reduce functions. Since Map Reduce is designed for processing a single input, the support of joining that requires more than two inputs with Map Reduce has been an open issue. We roughly classify join methods within Map Reduce into two groups: Map-side join and Reduce-side join.

b) Map-side join:

Map-Merge join is a common map-side joins that works similarly to sort-merge join in DBMS. Map-Merge join performs in two steps. First, two input relations are partitioned and sorted on the join keys. Broadcast join is

another map-side join method, which is applicable when the size of one relation is small. The smaller relation is broadcast to each mapper and kept in memory.

c) Reduce-side join:

Repartition join is the most general reduce-side join. Each mapper tags each row of two relations to identify which relation the row comes from. After that, rows of which keys have the same key value are copied to the same reducer during shuffling.

VI. APPLICATIONS

Map Reduce is simple and effective for computing aggregate. Therefore, it is a great deal compared with "filtering then group-by aggregation" query processing in a DBMS. Here are major advantages of the Map Reduce framework for information processing. Simple and easy to use The Map Reduce model is simple but expressive. With MapReduce, a programmer defines his job with only Map and Reduce functions, without causing to specify physical distribution of his job across nodes. Flexible Map Reduce does not have any dependency on data model and schema.

VII. FRAMEWORK

Joiners and reducers actually run inside the same reduce task. An alternative that runs Map- Join-Reduce with two consecutive MR jobs is also proposed to avoid modifying the Map Reduce framework. For multi-way join, join chains are represented as a left-deep tree. Then previous joiners transfer joined tuples to the next joiner that is the parent operation of the previous joiners in the tree.

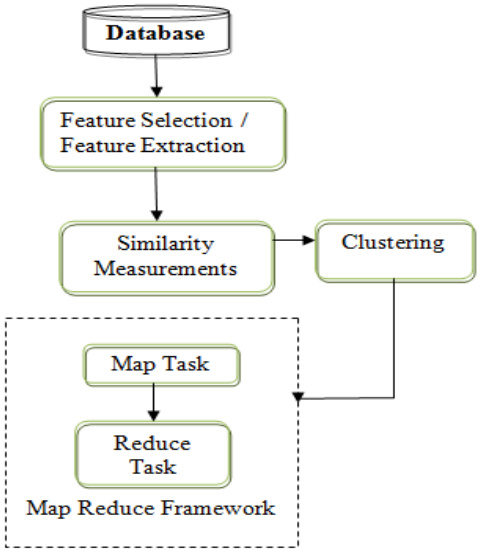

Fig.4. System framework

VIII. CONCLUSION

We have demonstrated how Map Reduce can be employed to implement various data clusters. In the process, we also demonstrated how Map Reduce frameworks can collaborate with Database Management Systems allowing for interesting possibilities. Low efficiency with fault-tolerance and scalability as its principal goals, Map Reduce operations are not always optimized for I/O efficient. Map Reduce is a viable answer to processing problems involving large quantities of information. Especially for problems that can easily be partitioned into independent sub tasks that can be worked out. In the hereafter, it is viable that the need for cluster computing will grow as problem sets become larger and more interest gets in the area. Map Reduce will undoubtedly become a more popular solution and much used paradigm. We talked about the implementation part of Map Reduce and classified its improvements. Map Reduce is simple but offers better scalability and fault-tolerance for massive information processing. However, Map Reduce is unlikely to substitute DBMS, even for data warehousing.

ACKNOWLEDGMENT

We would like to thank everyone who has motivated and supported us for preparing this manuscript.

REFERENCES

[1] Qinbao Song, Jingjie Ni and Guangtao Wang, A Fast Clustering-Based Feature Subset Selection Algorithm for High Dimensional Data, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING VOL:25 NO:1 YEAR 2013
[2] Indranil Palit and Chandan K. Reddy, Member, IEEE, Scalable and Parallel Boosting with MapReduce, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 10, OCTOBER 2012
[3] Makho Ngazimbi, DATA CLUSTERING USING MAPREDUCE, March 2009.
[4] TARUN DHAR DIWAN, PRADEEP CHOUKSEY, R. S. THAKUR & BHARAT LODHI, Exploiting Data Mining Techniques For Improving the Efficiency of Time Series Data, BIRT, Bhopal M.P. India
[5] Alina Ene, Sungjin Im, Benjamin Moseley, Fast Clustering using MapReduce.
[6] Elena Tsiporkova,Veselka Boeva,Elena Kostadinova, MapReduce and FCA Approach for Clustering of Multiple-Experiment Data
[7] Compendium, Technical University of Sofia-branch Plovdiv, Plovdiv, Bulgaria.
[8] Robson L. F. Cordeiro, Caetano Traina Jr, Agma J. M. Traina, Clustering Very Large Multi-dimensional Datasets with MapReduce,San Diego, California, USA.Copyright 2011
[9] Sun Zhanquan, Geoffrey Fox, A Parallel Clustering Method Study Based on MapReduce, School of Informatics and Computing, Pervasive Technology Institute, Indiana University Bloomington, Bloomington, Indiana, 47408, USA)
[10] Junbo Zhang, Dong Xiang, Tianrui Li, and Yi Pan, M2M: A Simple Matlab-to-MapReduce Translator for Cloud Computing, TSINGHUA SCIENCE AND TECHNOLOGY IS SN 1 0 07 - 0 2 14 0 1 / 1 2 p p 1-9 Volume 18, Number 1, February 2013
[11] Kyong-Ha Lee, Hyunsik Choi, Bongki Moon, Parallel Data Processing with MapReduce: A Survey, SIGMOD Record, December 2011 (Vol. 40, No. 4)
[12] Ralf L¨ammel, Google's MapReduce Programming Model,Data Programmability Team Microsoft Corp., Redmond, WA, USA
[13] Jeffrey Dean and Sanjay Ghemawat, MapReduce: Simpli_ed Data Processing on Large Clusters, January 2008/Vol. 51, No. 1

Authors

Puppala Priyanka[1] is pursuing her Post Graduate in Master of Technology with specialization of Computer Science & Engg. at AVN Inst. of Engg.& Tech, Hyderabad, AP, India. Her interested research area is Data warehousing & Data Mining, Network Security and Data Structures.

Prof. Shaik.Abdul Nabi[2] is the Head of the Dept. of CSE, AVN Inst.Of Engg.& Tech, Hyderabad, AP, India. He completed his B.E (Computer Science) from Osmania University, A.P. He received his M.Tech. from JNTU Hyderabad campus and now he was submitted his Ph.D. thesis in the area of Web Mining from Acharya Nagarjuna University, Guntur, AP, India. He is a certified professional by Microsoft. His expertise areas are Data warehousing and Data Mining, Data Structures & UNIX Networking Programming.

Meena Kumari P[3] is pursuing her Post Graduate in Master of Technology with specialization of Computer Science & Engg. at AVN Inst. of Engg.& Tech, Hyderabad, AP, India. Her interested research area is Data warehousing & Data Mining, Database Security and Data Structures

