An Efficient Keyword Based Search of Big Data Using Map Reduce

P. Srinivasa Rao
Department of CSE, MVGRCE, Vizianagaram

M. H. M. Krishna Prasad
Department of CSE, JNTUK, Kakinada
Email:krishnaprasad.mhm@gmail.com

K. Thammi Reddy
Department of CSE, GITAM University, Visakhapatnam
Email:tammireddy@yahoo.com

***Abstract*—With the arrival of the data deluge, traditional and centralized tools used to extract knowledge from data become obsolete due to their limited ability to handle massive data. To cope with the need for scalable solutions, a new framework has emerged: Hadoop, an open-source ecosystem designed for storage and large-scale processing work on a cluster of commodity hardware. In order to overcome the limitations in key word based information retrieval systems, an efficient methodology has been designed. A system with the new approach mimics the real world, where every task is laced with certain indexing as this is basic idea behind knowledge processing. Hadoop and R: open source frame works for storing and processing large datasets, are used for preprocessing the text documents. First, a set of text documents are considered. Preprocessing is performed on a large domain of data using R. This includes the removal of the stop words along with stemming and excluding less frequency words. Despite this preprocessing, owing to the colossal number of index terms still floating in the considered domain data, the problem of high dimensionality is encountered. Therefore the dimensionality of such a group of terms is reduced by incorporating a keyword based methodology in Hadoop MapReduce Framework. The developed Model is useful for processing the query which gives us the relevant information with low response time from the data pool considered.**

***Index Terms*—Hadoop, MapReduce, Bigdata, HDFS, information retrieval systems**

I. Introduction

Information retrieval is the science of searching for information in a document, searching for document themselves, searching for metadata that describe data and for data bases such as text, image or sound. The reason for the need of information retrieval is to process large corpus (the group of documents over which we perform retrieval is called collection or corpus) quickly, to allow more flexible searching operations, to allow ranked retrieval. Presently many search engines and systems are based on keyword based retrieval methodology which has its own limitations regardless of many effective improvements in its retrieving methods. Keyword based retrieval systems face difficulties in conceptualization of user needs. Aiming to solve the limitations of keyword-based models,a new approach has been introduced in the paper to solve the problems in the Information Retrieval(IR) [1]. At the core of these new technologies, Hadoop were envisioned as key elements to Process massive data in applications related to key word based search [2]. Modern information retrieval systems need the capability to reason about the knowledge conveyed by text bases. The Model framework constructed [3] allows users to query the accurate content of the documents. The Corpus considered is a collection of concepts and their interrelationships [4] which can collectively provide an abstract view of an application domain**.**

The preprocessing of documents can be done by using an open source tool called R. Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

The retrieval of huge information can be improved using one powerful tool called Hadoop MapReduce. Representation of document as a document term vector leads to high dimensionality problem as there will be enormous number of index terms in corpus. It is possible to reduce the dimensionality by considering equivalence classes of terms associated with related concepts as single dimension for rough set model. So clustering is used to group the terms. A cluster is described as a set of similar objects or entities collected or grouped together. All entities within a cluster are alike and the entities in different clusters are not alike. Each entity may have multiple attributes, or features and the likeness of entities

Manuscript received December 21, 2016; revised April 1, 2017.

doi: 10.12720/jait.8.3.159-164

is measured based on the closeness of their features. Therefore, the crucial point is to define proximity and a method to measure it. There are many clustering techniques and algorithms in use. K-means is the most common and often used algorithm. K-means algorithm takes an input parameter k, and partitions a set of *n* objects into *k* clusters according to a similarity measure.

However the results of partition clusters are highly influenced by the initial selection of seed points. It may not generate quality of clusters if it starts with randomly generated seed points. One solution is to make use of multiple set of random seed points as the basis for generation of clusters and selecting the best of clusters based on their quality. This requires a lot of computational resources. We have many paths in which we can carry out all the required computations where hadoop is one among those paths available which is fast robust easier to understand and relatively efficient to perform the required computations. Hadoop whose system model is shown in Fig. 1 is extremely scalable. Major component of Hadoop HDFS (for storage) is optimized for high through put. Hadoop is an open source software framework that can run large data-intensive, distributed applications and can be installed on commodity Linux clusters. Hadoop comes with its ownfile system called the Hadoop Distributed File System (HDFS) and a strong infrastructural support for managing and processing huge petabytes of data. Each HDFS cluster consists of one unique server called the Name node that manages the namespace of the system, determines the mapping of blocks to Data nodes, and regulates file access. Each node in the HDFS cluster is a Data node that manages the storage attached to it. The data nodes are responsible for serving read and write requests from the clients and performing block creation, deletion and replication instructions from the Name node.

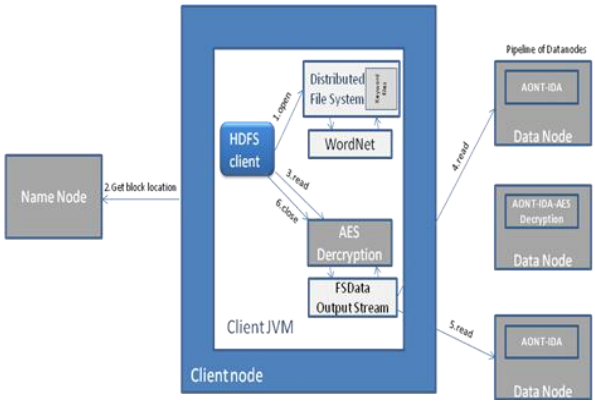

Figure 1. System model using Hadoop MapReduce.

II. Related Work

The theoretical foundations, research and applications of Formal Concept analysis (FCA) in different areas has been described in the paper by A.K.Sarmah, S.M.Hazarika and S.K.Sinha [5]. It stated that Formal Concept Analysis started off as a field of applied mathematics based on applied mathematical order theory that tries to achieve a way of explicitly representing human conceptual thinking. Here, they told that concepts are represented as conceptual hierarchies which allows the analysis of complex structures and discovers the dependencies within data. It stated that research and usage of FCA has been in different areas since the pioneering work done during the early 1980s. Initially it started to be used in the area of Knowledge Representation and Reasoning (KR&R). Applications of FCA in the area of KR&R has been chiefly in ontology. It also described Lattice theory where lattice is defined as a partially ordered set (*poset*) in which every pair of elements has a unique supremum or Lowest Upper Bound (LUB), called their JOIN and an infimum or Greatest Lower Bound (GLB), called their MEET. It also stated that Ontology Engineering is a prime domain where FCA has been used for the purpose of concept classification. But the limitations are FCA tools and techniques require data or the context to be provided in discrete form. One of the future directions of research in this regard would be to incorporate parsing techniques into FCA tools so that data presented in other forms like continuous text, expressions etc. could be explored for concepts. Visualization and generation of the concept lattice can be done conveniently by the existing algorithms for a limited and small context. However, as the size of the context increases, this becomes an issue.

Thomas C.Jepsen [6] describes some definitions of "ontology" as it relates to computer applications and gives an overview of some common ontology-based applications. From a more modern perspective, ontologies came to be of interest to computer scientists in the 1970s as they began to develop the field of artificial intelligence. They realized that if you could create a domain of knowledge and establish formal relationships among the items of knowledge in the domain, you could perform certain types of automated reasoning. Tom Gruber, a computer science scientist introduced the term in his paper in 1993. Thomas also described the properties of ontology, types of ontology and also how is it different from hierarchies.

Marek Obitko, *et al.* [7] described how to design Ontology using Formal concept analysis. Their ontology design allows for discovering necessity for new concepts and relations, which leads to an ontology which is suitable for knowledge exchange and information retrieval. The main characteristics of their method are: concepts are described by properties. The properties determine the hierarchy of concepts. When the properties of different concepts are same, then the concepts are same as well. However if the context increased, construction and navigation of ontology becomes complex.

Suraj [8] stated that rough set theory is mathematical approach to imperfect knowledge and recently it has become crucial issue for computer scientists he explained the advantages of rough sets.

Jurka [9] described how rough sets can be used in process of data mining from databases. The object of this work was to suggest the possibilities of alternative methods of data mining. It is about the rough sets theory

and its use for the mining of decision rules data.It is advantageous for mining of incomplete data.

Rough sets and information retrieval is a paper by Padmini Das Gupta [10], which applies rough sets to the design of information retrieval accessing collection of documents. Advantages offered are term weighting and ranking of retrieved documents.

K.Thammi Reddy, *et al*. [11] proposed a rough set based information retrieval in which they presented a hybrid clustering approach for the formation of equivalence classes of terms associated with related concepts. They also proposed a new term weight estimate namely term probability–inverse document frequency (TF-IDF) for representing a term as a vector before clustering the terms. Clustering is performed to group together related terms of a concept into equivalence classes, which can be used to reduce the dimensionality of the documents for rough classification.

Yinghui Huang, *et al*. [12] described generation of rough ontology. They stated that rough ontology is an extension of ontology, in which precise concepts and precise relation between the concepts is described. Rough ontology is a kind of acceptation of the uncertainty about the world and is a basic tool of knowledge processing and knowledge application. They proposed a method which generates rough concept lattice from a context by using increment, and then this concept lattice is converted to rough concept tree by clustering. Then this tree is mapped to ontology. The limitations of this are it becomes complex when applied to large data sets.

Latifur Khan, *et al*. [13], described ontology generation from documents. The traditional search mechanism employs key word based search. So they proposed a design and implementation of domain dependent ontology which can be used in semantic based information retrieval. For this they used various clustering algorithms for construction of hierarchy and then to find an appropriate concept for each node in the hierarchy they used assignment algorithm. However this paper did not consider the uncertainty in the data.

III. METHODOLOGY

In the big data era, it is very difficult to even view the data because of its size and other parameters. So to process such data which has horizontal scaling property traditional tools are not suitable so modern bigdata tools have to be utilized. So in the paper hadoop framework with mapreduce functionality is developed to retrieve the documents of interest from the pool.

The generation of Ranked documents from Documents pool is highlighted in Fig. 2.

Basic Operations

1. Segmentation: The segmentation is a step used to isolate each term of the corpus and also to filter the terms used in the text analysis. It is done according to punctuation. Then conversion of uppercase to lowercase. Finally filtering of words with at least 3 and at most 50 characters.

2. Stop-Word Elimination: It is the process of eliminating the inessential data in the document. In the 8 parts of speech available, those which fall under stop-words are pronouns, prepositions, conjunctions and interjections. Therefore, the mentioned parts of speech are eliminated from the document.

3. Word Stemming: A stemming algorithm is a process of linguistic normalization, in which the variant forms of a word are reduced to a common form, for example; connections, connected, connecting, connective etc. are variant forms of the word "Connect".

4. Word Count: To identify the Term-frequency the words are counted based on number of repetitions. The required output is achieved here itself but in a whole and sole document.

5. Document Clustering: Clustering algorithms in computational text analysis group's documents into what are called subsets or clusters where the algorithm's goal is to create internally coherent clusters that are distinct from one another.

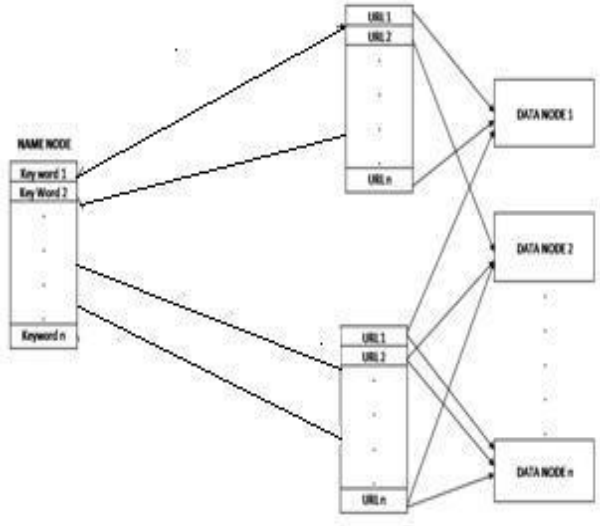

Figure 2. Mapping of keywords by using Hadoop MapReduce

Retrieving the documents related to user query using the Model is highlighted in Fig. 3. The query passage is tokenized and stop words are removed. Words appearing in different morphological forms are mapped on to their common terms. Frequency of each term is calculated and least useful words are stripped. Synonyms are attached to each word present. The clustered query is mapped on to Model constructed and relevant documents are ranked and retrieved. The documents are displayed to the user.

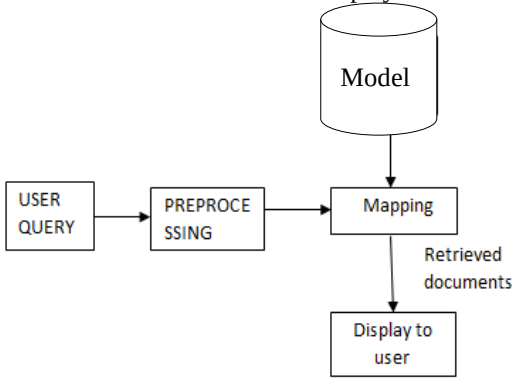

Figure 3. Mapping of keywords with developed Model

The detailed algorithms are presented in the following sections.

A. Preprocessing

Documents containing repeated occurrences of keywords, not all the words of a document are to undergo preprocessing in order to derive a well-defined index terms. In this process the documents are made to undergo a series of transformations like removing the Stop words,

Stemming is a process of linguistic normalization, in which the variant forms of a word are reduced to a common form. This process is done to the tokenization output. Frequently, the performance of an information retrieval system will be enhanced if term groups such as this are changed into a single term.Porter stem Algorithm is used based on the idea of suffixes.

The importance of the document associated with the keyword is determined by the frequency of keyword in a document. The keywords with least frequency as well as the keywords that occur very often in most of the documents are less likely to be significant in finding the weight of a document. To strip the least useful words we first calculate the corpus frequency i.e., the number of documents containing the term using Hadoop MapReduce and then those with least frequency and highest frequency are removed.

B. Clustering

We make over the preprocessed data into term-document matrices, each of which limited to the collection of documents belonging to a selected corpus. This can be used in grouping the similar terms. Clustering methods are used to identify the groups of terms based on their similarity estimates. We can use tf-idf estimate of a term in a document as a component of the term vector. Tf-idf estimates is given by:

$$\text{TF-IDF} = tf_{ij}/n*\log(m/df_i)$$

$tf_{ij}$is the term frequency of ith term in jth document
n is number of words in jth document
m is total number of documents
df_i number of documents in which ith term appears

A cluster is described as a set of similar objects or entities collected or grouped together. All entities within a cluster are alike and the entities in different clusters are not alike. Each entity may have multiple attributes, or features and the likeness of entities is measured based on the closeness of their features. Therefore, the crucial point is to define proximity and a method to measure it. To estimate the proximity of terms we use Euclidean distance. The distance between two things (a & b) using Euclidean distance is given by:

$$d(a,b) = SqRt\ [\ \sum_{i=1}^{n}(bi - ai)2\ ]$$

For calculation of each of these term frequency, total number of words in a document, number of documents containing the term and finally TF-IDF is done using Hadoop MapRreduce.

In K-means algorithm, k stands for the number of clusters (groups) to be formed, hence this algorithm can be used to group known number of groups within the analysed data.K Means is an iterative algorithm and it has two steps. First is a Cluster Assignment Step, and second is a Move Centroid Step.

CLUSTER ASSIGNMENT STEP: In this step, we randomly chose two cluster points (red dot & green dot) and we assign each data point to one of the two cluster points whichever is closer to it.

MOVE CENTROID STEP: In this step, we take the average of the points of all the examples in each group and move the Centroid to the new position i.e. mean position calculated.

The above steps are repeated until all the data points are grouped into k groups and the mean of the data points at the end of Move Centroid Step doesn't change

In this phase the query passage is tokenized and stop words are removed. Stemming is performed on these words and is mapped on to their common stems. This list of stems in the query passage obtained is mapped with reference to the k equivalence classes obtained during generation of relevant documents.

Now for a given keyword to search the file in which the the keyword is most frequently occurred. Identify the documents which are having more relevance to the given user query. If multiple documents exists in this group rank them based on their similarity to the query and is represented in Table I.

TABLE I. CALCULATING RELEVANCE SCORE FOR THE WORD TO RETRIEVE DOCUMENTS.

Word	W_i				
File ID	F_{i1}	F_{i2}	F_{i3}	….	F_{iNi}
Relevance Score	0.00465	0.0125	0.00943	0.07643	0.09573

IV. EXPERIMENTATION

A. Environment Setup

The experiments were performed on a 4 node cluster equipped with Hadoop. This project has been provisioned with one Name Node and four Data Nodes. The Name Node was configured to use two 2.5-GHz CPUs, 2 GB of RAM, and 500 GB of storage space. Each Data Node was configured to use two 2.5-GHz CPUs, 2 GB of RAM, and 500 GB of disk storage. Besides this, all the computing nodes were connected by a gigabit switch. BOSS GNU Linux 4.1., Hadoop 0.20.1, and Java 1.6.0_6 were installed on both the Name Node and the Data Nodes.

The evaluation of IR model is carried out by using the objective retrieval quality testing methodologies. The documents retrieved for the given user preference keyword are evaluated by using metrics such as

Precision (P), Recall (R), F-measure (F), Error Rate (E) ,Accuracy (A) for developed (SBIRS) and existing Key based Information Retrieval (KBIRS) model

For experimentation we have taken 400,800 and 1000 documents from corpus data set and the observed results

are drawn in Fig. 4 in which the metric Precision is considered. In Fig. 5 the metric Recall is used. In Fig. 6 F-measure is used and finally in Fig. 7 Accuracy and Error rate are considered to evaluate the performance of the framework developed.

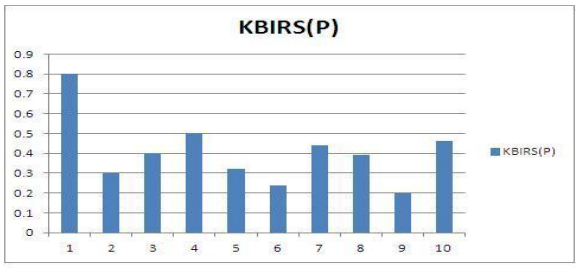

Figure 4. Precision versus number of documents in HDFS.

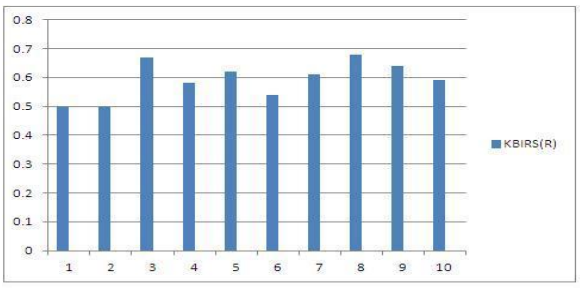

Figure 5. Recall versus number of documents in HDFS.

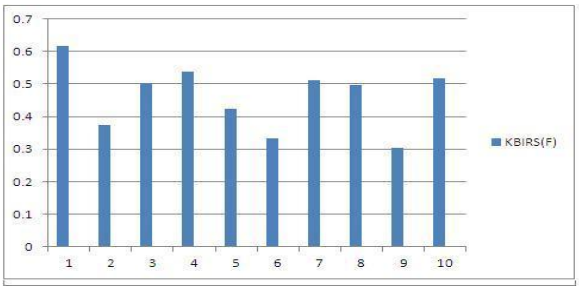

Figure 6. F-Measure versus number of documents in HDFS.

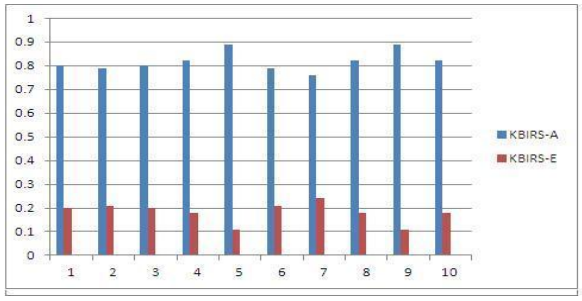

Figure 7. Accuracy versus Error rate

V. Conclusions And Future Work

The paper covers the general overview of the text processing system, involving different steps like Segmentation, Stop-word Elimination, Stemming, Word count and Clustering to analyze unstructured data to retrieve the document of user interest.

The main claim behind the implementation is how techniques are developed to enhance the effectiveness of processing a given text file. By this we can analyze different kinds of data like twitter tweets, Wikipedia articles, blog comments, etc. We have shown how the data is obtained after the final step in the form of clusters and evaluated using different quality metrics. The experimental results of the works are also presented in order to make a comparison. It is observed that the presented work performs one of the best up until now in terms of text mining.

At present the paper satisfies the major functionality i.e., to mine the information from unstructured data to retrieve the documents from the pool. A plan is to set up a graphical user interface with semantics to reduce human interaction and to get accurate results. Another bold step is to make the project accept any kind of input data like data in the form of 'PDF', 'Word document', 'XML' etc.

The framework can be deployed in a cloud environment to provide computation as a service on other real or benchmarked data sets.

It is also better to provide service as an object in which user facial features or other biometrics can be taken once again before providing service to user, to avoid service misuse among authorized users.

The Information Retrieval application deployed in the extended framework can also be used to support multiple keywords search with negation words. New approaches can also be designed for IDF factor to preserve the order while calculating the score for the keywords provided by the user.

The above issues may be considered for future task in the overall development of Key based ranking encrypted data systems.

References

[1] A. M. Al-Zoghby, A. S. E. Ahmed, and T. T. Hamza, "Arabic semantic web applications–A Survey," *Journal of Emerging Technologies in Web Intelligence*, vol. 5, no. 1, 2013.

[2] S. Menaka and N. Radha, "Text classification using keyword extraction technique," *IJARCSSE*, vol. 3, no. 12, 2013.

[3] V. B. Mititelu, "Increasing the effectiveness of the Romanian wordnet in NLP applications," *Computer Science Journal of Moldova*, vol. 21, no. 3, 2013.

[4] L. L. Meng and J. Z. Gu, "A new method for calculating word sense similarity in WordNet," *International Journal of Signal Processing*, vol. 5, no. 3, 2012.

[5] P. U. bulakh, et al, "Application of semantic similarity using ontology for document comparison," IJRCM, vol. 3, no. 12, 2013 .

[6] T. C. Jepsen, "Just what is a ontology, anyway?" *IT Professional*, vol. 11, no. 5, pp. 22-27, 2009.

[7] M. Obitko, V. Snasel and J. Smid, *Ontology Design with Formal Analysis*, *Communications in Computing*, pp. 302-310, 2014.

[8] A. Amine, et al, "Evaluation of text clustering methods using WordNet," *The International Arab Journal of Information Technology*, no. 7, no. 4, 2010.

[9] M. Zurini, "Word sense disambiguation using aggregated similarity based on WordNet graph representation," *Informatica Economică*, no. 17, no. 3, 2013.

[10] C. Bouras and V. Tsogkas, "A clustering technique for news articles using WordNet," *Elsevier*, vol. 36, pp. 115-128, 2012.

[11] K. T. Reddy, M. Shashi, L. P. Reddy, "Hybrid clustering approach for term partitioning in document data sets," Artificial Intelligence and Pattern Recognition, pp. 165-172, 2007.

[12] J. B. Gao, B. W. Zhang, X. H. Chen, "A WordNet -based semantic similarity measurement combining edge- counting and information content theory," *Elsevier*, vol. 39, pp. 80-88, 2014.
[13] S. Vijay, "A comparison of different measures to evaluate the semantic relatedness of text and its application," *IJRTE* , vol. 1, no. 1, 2012.

Dr. P.Srinivasa Rao currently working as an Associate Professor in the Dept.of CSE, MVGR College of Engineering. He is having Over 12 years of teaching experience. His research includes Data Analytics, Cloud Computing, Data Mining, Distributed Computing, Data analytics Image Processing etc.

Dr. Munaga HM Krishna Prasad is currently Full Professor, Department of Computer Science and Engineering, University College of Engineering Kakinada (Autonomous), JNTUK, Andhra Pradesh. He did his B.E. from Osmania University, Hyderabad, M.Tech. and Ph.D. Computer Science and Engineering from JNTU, Hyderabad. DrMunaga successfully completed a two year MIUR fellowship (Jan 2007 – Dec 08) at University of Udine, Udine, Italy. He has about 50+ research papers in various International Journals and Conferences, and attended many national and international conferences in India and abroad. He is a member of Association for Computing Machinery (ACM), ISTE and IAENG (Germany) is an active member of the board of reviewers in various International Journals and Conferences. His research interests include data mining, BigData Analytics and High Performance Computing.

Dr. K. Thammi Reddy is the Director of Internal Quality Control(IQC) and Professor of CSE. at Gandhi Institute of Technology(GITAM).He is having Over 18 years of experience In teaching, Research,Curriculam Design and consultancy. His research areas include Data warehousing and Mining, Distributed computing, Network Security etc.