GA Based Model for Web Content Mining

Vikrant Sabnis, R. S. Thakur
Department of Computer Applications, MANIT, Bhopal, India

Abstract

Several methods are available for mining frequent patterns in web data, but mostly they suffer from the problem of huge candidate generation and number of database scans. In view of above a genetic based model for mining frequent patterns in web content data. In the proposed genetic operator, crossing over method leads to offspring which must survive the certain fitness test or conditions to become frequent pattern and ancestor for next patterns. In this way the useless individuals or candidates are pruned out thereby reducing the number of candidates for next test. Also this approach requires only one scan of database. Thus this model is able to address the issues of large number of candidate generation and number of database scans.

Keywords: *Genetic Algorithm, Mutation, Crossing over, Population, Web content mining*.

1. Introduction

Data mining involves the study of data-driven techniques to discover and model hidden patterns in large volumes of raw data [1]. The application of data mining techniques to Web data is referred to as Web data mining. Web mining is the application of data mining techniques to extract knowledge from web data, i.e. web content, web structure, and web usage data. The process may involve preprocessing the original data, integrating data from multiple sources, and transforming the integrated data into a form suitable for input into specific data mining operations [2],[3],[4],[5]. Web content mining involves efficiently extracting useful and relevant information from different web sites and databases.

Web content mining targets the knowledge discovery, in which the main objects are the traditional collections of text documents and also the collections of multimedia documents such as images, videos, audios, which are embedded in or linked to the Web pages [6][7][8][9].

Mining frequent patterns in dataset is a field of data mining that has received a lot of attention in recent years. There are number of ARM approaches reported in the literature [10][11][12][13][14]. But these traditional approaches suffer from the problem of number of database scans and candidate generation. Some attempts are reported in the literature to address these issues [15][16][17][18] [19][20][21].

2. Related Work

R. Vijaya Prakash et. al. [22] has proposed the application of Genetic Algorithm (GA) for improvement of the generation of Frequent Item set with numeric attributes instead of binary or discrete attributes. They have claimed that this approach will be advantageous in the discovery of frequent item sets during global search with relatively less time due to greedy approach.

For extracting useful information from very large databases Peter P. et. al. [23] have formulated a generalized association rule mining model with a combination of genetic algorithms and a modified a-priori based algorithm. Their model maps association rule mining problem as a multi-objective combinatorial problem using genetic algorithm. The combination of genetic algorithm with a-priori query optimization yields faster results.

Minh Nghia Le et. al [24] presented frequent Schemas Analysis (FSA) of Optinformatics, which uses Binary GA to mine implicit frequent schemas. Their proposed mining algorithm mines interesting frequent schemas that exist implicitly within the archived Binary GA data based on pruning criteria. Algorithm significantly performs well over the naive FP-growth approach.

G. Vijay Bhasker K. et. al. [25] proposed comparative study between apriori algorithm with genetic algorithm. The comparision has been done using 32 items, 10 transactions and found that execution time of Genetic algorithm is much better than the apriori algorithm.

Xiaowei Yan a et. al. [26] designed a genetic algorithm-based strategy for identifying association rules without specifying actual minimum support their work was based on elaborate encoding method and for fitness function they used relative confidence.

3. Genetic Algorithm and Its Process

Genetic algorithm (GA) [27] was first introduced by John Holland in 1975. This is based on evolution theory (natural selection and genetics) that employs optimization of search space. In other words it is an evolutionary computational strategy, because it is based on natural evolution selection of population on basis of their survivability. Survivability

of population depends on the fitness constraint of environment (survival of fittest). GA is probabilistic search procedure using some genetic operators to a set of variables or genes in problem space. Environment defines the fitness of population in problem space. If a population supports minimum fitness in that environment condition, population survives and is included in population for further reproduction, otherwise population is rejected. In GA variables are represented by chromosomes. Chromosomes have set of genes, Gene is elementary unit of heredity and it is represented in two alternate forms i.e. alleles. One form is dominant character and another recessive character.

Chromosomes cross over each other to produce offspring (or Progeny). This process is called recombination. Alleles are responsible for distribution of traits and observation of traits i.e. inherited in pair one from each parents. Alleles are responsible for survivability of offspring. On the basis of alleles we can analyze the mutation in individuals or offspring. The spontaneous change in individuals to produce unexpected traits is called mutation. It is rare in population. There are some factors that cause mutation and that factor is called mutant. Mutation may be sometime harm full for individual, traits or beneficiary for survivability, because mutation may increase or decrease fitness of individuals. Mutation may be helpful for optimization and searching solution in problem space [28].

Genetic Algorithms process is a three step searching goal state in problem space [28].

1- Initial population/ Reproduction:

Set of Variables are represented by chromosome or individuals. Variable may be Binary encoded form or real value etc. and each individuals associated with fitness value. On the basis of minimum fitness value or constraint, the population will be selected for reproduction.

2- Crossing Over:

Select two individuals from population randomly or on the basis of fitness value (Genetics theory healthy parents probable to produce healthy Childs). Cross over each other to produce offspring (progeny).

3-Mutation:

Define probability of mutation inverted bit of off spring to improve fitness of individuals. The mutation may occur and may not occur in individuals. Mutation happens in a rare case. Calculate fitness value of offspring. If offspring satisfies minimum fitness value, select the individuals for further reproduction.

Repeat Crossing over and Mutation until we get the complete solution.

4. Frequent Pattern Mining in Web Content Database using GA

The Apriori [10] and other traditional pattern mining approaches [11][12][13][14] generate many candidates which leads to increase in memory requirements and larger number of database scans. All these approaches use transactional datasets. Here GA based model is proposed for frequent pattern mining in web content database using various genetic operators (Reproduction, Crossover, mutation). This algorithm uses minimum threshold fitness as measure for selecting contents (items) as frequent. Web contents are represented as chromosomes or individuals and sessions are represented as alleles. Alleles are represented on chromosomes in the form of binary codes (1= dominants 0= recessive). In a chromosomes 1 indicate that content is visited in particular session and 0 indicate that content is not visited in a particular session. This is illustrated in the fig.1 below:

Session	Set of contents		S1	S2	S3	S4
S1	{C1, C2, C3}	ChromosomeC1=	1	1	1	0
S2	{C1, C3}	ChromosomeC2=	1	0	1	1
S3	{C1, C2}	ChromosomeC3=	1	1	0	1
S4	{C2, C3}					

Fig.1: Binary representation of alleles in chromosomes

Steps of Genetic algorithm for finding frequent contents:

1- Perform binary encoding of session and contents in the form of chromosome to create population.

2- Select initial population (all chromosomes-C) on the basis of minimum threshold fitness.

3- Select two chromosomes-C as parent from initial population and perform arithmetic crossover (logical AND) to produce offspring. Because of arithmetic crossing over only common alleles are transferred in offspring,

4- In Arithmetic crossing over, Possibility of mutation is 0%. Perform inspection of mutation. Inverting bit of offspring is performed i.e. 1 is replaced with 0 or vice versa in case of mutation.

5- Calculate threshold fitness for offspring. If offspring satisfies minimum threshold then select both parent chromosomes-C as frequent and all associated ancestor chromosomes are frequent and include offspring in population for reproduction.

6- If selected two parent chromosomes-C crossover with their parent (Sibling) to produce offspring and if offspring satisfies minimum threshold fitness then select all subsets of parent chromosome-C as frequent.

7-Repeats steps 3,4,5, until all possible frequent chromosome-C (content sets) are found or all population generated satisfies minimum threshold fitness.

Genetic Operation to find surviving population (frequent content sets) of chromosome-C

1- Selection of Initial population or Reproduction:

In web content dataset which have contents, they appear in a particular session (or how many objects are accessed). Binary encoding of each content set represents its access (visit) in the corresponding sessions. Define fitness function f(x) for the chromosome-C.

Fitness Function:

Fitness function is a mathematical function to calculate the survivability of chromosome. Calculate percent support probability ratio of each chromosome-C or individual. Here items are treated as chromosome in the form of binary code.

Where

1= indicates item sold in a particular session.

0= indicates item not sold in a particular session.

The Fitness function support is given by

$$F(s) = \frac{\textbf{Total Number of true bit in chromosome} - \textbf{C}}{\textbf{Total length of chromosome} - \textbf{C}}$$

F(s)= TDAc / TLc

Where:

TDAc= Total dominants alleles in chromosome-C

TLc= Total length of chromosome-C

Percent support f(Sp)= f(s)*100= TDAc/TLc*100.

Relative percentage of each Individual is :

$$F(Rsp) = f(Sp) \Big/ \sum_{i=0}^{n} f(Spi)$$

Using above distributive measures fitness function evaluate fitness support of each individual in population. Compare the each individual's threshold value with user defined minimum fitness and Select all individuals as a initial population on the basis of user defined minimum threshold fitness support value for survivability of individual in environment (problem space) for reproduction. We have considered web content dataset D with content sets C= {C1, C2, C3, C4, C5, C6, C7} and 10 sessions which are shown in Fig.2.

Session	Content set
S1	{C1,C2,C5,C6}
S2	{C2,C3,C6}
S3	{C1,C2,C4}
S4	{C2,C3,C7}
S5	{C1,C2,C4,C6}
S6	{C1,C3}
S7	{C2,C3}
S8	{C1,C3,C6}
S9	{C1,C2,C3,C6}
S10	{C1,C2,C3,C5}

Fig. 2: Dataset D

There are ten chromosomes (content set) in dataset D, their binary encoding are shown by Fig 3.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Chromosome-C1	1	0	1	0	1	1	0	1	1	1
Chromosome-C2	1	1	1	1	1	0	1	0	1	1
Chromosome-C3	0	1	0	1	0	1	1	1	1	1
Chromosome-C4	0	0	1	0	1	0	0	0	0	0
Chromosome-C5	1	0	0	0	0	0	0	0	0	1
Chromosome-C6	1	1	0	0	1	0	0	1	1	0
Chromosome-C7	0	0	0	1	0	0	0	0	0	0

Fig. 3: Binary encoding of chromosomes (content set)

Evaluate fitness support percents of each Individual.

F(Sp)= (TDAc/TLC)*100

Chromosome-C1= (7/10)*100= 70%

Chromosome-C2= (8/10)*100= 80%

Chromosome-C3= (7/10)*100= 70%

Chromosome-C4= (2/10)*100= 20%

Chromosome-C5= (2/10)*100=20%

Chromosome-C6= (5/10)*100= 50%

Chromosome-C7= (1/10)*100=10%

We assume minimum support fitness= 20%

Individuals (chromosome-C7) do not satisfy the minimum support so that individual is not selected in initial population.

Then Initial population of individuals is:

Chromosome-C1	**1**	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**1**
Chromosome-C2	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**1**	**1**
Chromosome-C3	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**1**	**1**
Chromosome-C4	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
Chromosome-C5	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
Chromosome-C6	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**0**

According to our representation of chromosome we can select all individuals of initial population as frequent 1-content sets.

One set Frequent contents are = {C1, C2, C3, C4, C5, C6}

2- Crossing over (Recombination):

Choose two chromosomes from initial population on the basis of fitness value or randomly (copy some contents from one individual and other from another individuals) to produce full length of new offspring (child). There are number of ways of performing crossover like single point, two point, uniformly, and Arithmetic (AND operation) [29]. But here we are using arithmetic AND operation for crossing over two parents.

Calculate fitness value of offspring if child is survivable then select for further reproduction.

Step A:

Chromosome-C1 1 0 1 0 1 1 0 1 1 1
Chromosome-C2 1 1 1 1 1 0 1 0 1 1
AND arithmetic crossing over
Offspring (child) {C1, C2}c= 1 0 1 0 1 0 0 0 1 1

Fitness value of offspring= (5/10)*100=50%

Offspring satisfies minimum threshold fitness so both parents are selected as frequently associated.

Result A: Two set frequent content ={C1,C2}

If among selected two parents from population any of them or crossover with his parent of sibling and generated offspring (child) are survivable then all sub sets of ancestor are frequently associated with that individual.

Step B: Suppose we are performing crossover between {C1, C2}C chromosome and Chrosome-C3.

Offspring (child) {C1, C2}c= 1 0 1 0 1 0 0 0 1 1
Chromosome-C3 = 0 1 0 1 0 1 1 1 1 1
Arithmetic AND crossing over
Offspring (child) {C1, C2, C3}c= 0 0 0 0 0 0 0 0 1 1

Fitness value of Offspring (child) {C1, C2, C3}c =(2/10)*100=20%

Offspring {I1, I2, I3}c is survivable.

Result: B Three frequent content set = {C1, C2, C3}, as per property it's all two item sub sets are frequent= { {C1, C2},{C1,C3},{C2,C3}}.

Step C:

Suppose we select two individuals {C1, C2, C3}c×Chromosome-C4
Offspring (child) {C1, C2, C3}c= 0 0 0 0 0 0 0 0 1 1
Chromosome-C4 = 0 0 1 0 1 0 0 0 0 0
Arithmetic AND Crossing over
Offspring (child) {C1,C2, C3, C4}c = 0 0 0 0 0 0 0 0 0 0

Fitness of Offspring (child) {C1, C2, C3, C4}c = 0% . Thus offspring did not survive and 4-itemset {C1, C2, C3, C4} is not frequent.

Result C

It means chromosome-C4 may be may not be associated with some subset ancestor of partner individuals.

Step D:

Suppose we select two individuals {C1, C2}c × chromosome-C4
Offspring (child) {C1, C2}c= 1 0 1 0 1 0 0 0 1 1
Chromosome-C4 = 0 0 1 0 1 0 0 0 0 0
Arithmetic AND crossing over
Offspring (child) {C1, C2, C4}c= 0 0 1 0 1 0 0 0 0 0

Offspring (child) {C1, C2, C4}c fitness = (2/10)*100=20%

Offspring (child) {C1, C2, C4}c is survivable so we can select both parents as frequent items. And sub set associate ancestor also frequent.

Result D:

{C1, C2, C4} is three set frequent content and {C1, C4}, {C2, C4}, {C1, C2} as two set frequent content sets.

Step E: suppose we select Individuals {C1, C2}c×chromosome-C5
Offspring (child) {C1, C2}c= 1 0 1 0 1 0 0 0 1 1
Chromosome-C5 = 1 0 0 0 0 0 0 0 0 1
Arithmetic AND Crossing over
Offspring (child) {C1, C2, C5}c= 1 0 0 0 0 0 0 0 0 1

Fitness of (child) {C1, C2, C5}c= (2/10)*100=20%

Result E: Offspring is survivable and {C1, C2, C5} is a frequent three content set and its all subset are {C1, C5}, {C2, C5}, {C1, C2} frequent.

Step-F: Suppose we select two individuals {C1, C2}c × chromosome-C6
Offspring (child) {C1, C2}c= 1 0 1 0 1 0 0 0 1 1
Chromosome-C6 = 1 1 0 0 1 0 0 1 1 0
Arithmetic AND crossing over
Offspring (child) {C1, C2, C6}c= 1 0 0 0 1 0 0 0 1 0

Fitness of offspring (child) {C1, C2, C6}c= (3/10)*100=30%

Result F: Offspring is Survivable and {C1, C2, C6} is a Three set frequent contents and {C1, C6} {C2, C6}, {C1, C2} are two set frequent contents.

Step G: Suppose we select two chromosome-C3× Chromosome-C6
Chromosome-C3 = 0 1 0 1 0 1 1 1 1 1
Chromosome-C6 = 1 1 0 0 1 0 0 1 1 0
Arithmetic crossing over
Offspring {C3, C6}c= 0 1 0 0 0 0 0 1 1 0

Offspring (child) {C3, C6}c = (3/10)*100=30%

Result-G: Offspring is survivable and {C3, C6} is a two set frequent contents.

Note: Our selection process of population based on the fitness value and child parent sibling relationship. Because healthy parents produce probable healthy child and child cross over with parent sibling and offspring survives then all sub sets of ancestors of child are associated with parent sibling. We can also choose random population, but it will generate many useless crossing over operations.

3- Mutation Analysis:

Once crossing over was performed, Mutation may be take place. We can analyze mutation in binary encoded chromosome for inverting bit 'one =1' replace with 0 and vice versa.
Chromosome-C= 1100010000= 1000100000
Using mutation we can increase or decrease fitness of individual. It depends on mutant. But in the case of solving real problem when we analyze mutation operation it will depend on what are the goals of searching solution in problem space.
Arithmetic mutation transfers only common alleles (character) of parents to children. Survivability depends on the how many commons alleles are present in both parent chromosomes. In the arithmetic crossing over possibility of mutation is Zero percent.

4. Conclusions

This genetic algorithm based approach needs only one dataset scan and during scanning the datasets are converted into chromosome. It generates less number of candidates or offsprings and does not require level by level candidate generation done in traditional approaches. This is good approach to get maximal itemsets. It is easy to implement as parallel process to get frequent itemsets. In this case each pair of chromosomes can run on individual processor and perform logical AND operations. In next pass only surviving offsprings are used to repeat same process to get higher size offspring.
Further we can extend our work by generating the association rules from the frequent content set.

Acknowledgments

This work is supported by research grant from MANIT, Bhopal India under Grants in Aid Scheme 2010-11, No Dean(R&C)/2010/63 dated 31/08/2010.

References

[1] Margaret H. Dunham, "Data Mining Introductory & Advanced Topics", Pearson Education, 2006.

[2] J. Borges and M. Leavene. "Data mining of user navigation patterns". In proceedings of the WEBKDD'1999, Workshop on Web Usage Analysis and User Profiling, August 15, 1999, San Diego, CA, USA, pages 31-36, 1999.

[3] S. K. Madria, S. S. Bhowmick, W. K. Ng and E. P. Lim. Research issues in web data mining. In proceedings of Data Warehousing and Knowledge Discovery, First International Conference, DaWaK'99, Pages 303-312,1999.

[4] Qingyu Zhang and Richard S. Segall," Web mining: a survey of current research,Techniques, and software", in the International Journal of Information Technology & Decision Making Vol. 7, No. 4 (2008) 683–720.

[5] Kosala and Blockeel, "Web mining research: A survey," SIGKDD:SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery and Data Mining, ACM, Vol. 2, 2000.

[6] Xiaoshu Hang, Honghua Dai, Youhua Zhang, "An Web Content Based Data Mining for Car Consumption Preference in China" 0-7803-8242-0/03/ Q 2003 IEEE.

[7] Camelia Elena CIOLAC, Florica LUBAN, Răzvan Cătălin DOBREA, "Web Content Mining Framework for Discovering University Formations' Compatibility with the Market Needs" Review of International Comparative Management, Volume 11, Issue 5, December 2010.

[8] Alisa Kongthon, Niran Angkawattanawit, Chatchawal Sangkeettrakarn, Pornpimon Palingoon, Choochart Haruechaiyasak , "Using an Opinion Mining Approach to Exploit Web Content in Order to Improve Customer Relationship Management" 978-1-890843-21-0/10/2010 IEEE.

[9] Juan Vel´asquez, Hiroshi Yasuda and Terumasa Aoki, "Combining the web content and usage mining to understand the visitor behavior in a web site" Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03) 2003 IEEE.

[10] Agrawal R., Imielinski T., and Swami A. Mining association rules between sets of items in large databases. In Proc. SIGMOD, pages 207-216, May1993.

[11]Agrawal R. and Srikant R. Fast algorithms for mining association rules in large databases. In Proc. 20th VLDB, pages 478-499, Sept. 1994.

[12]Sarasere A.,Omiecinsky E., and Navathe S. An efficient algorithm for mining association rules in large databases. In Proc. 21st VLDB, pages 432-444, Sept. 1995.

[13]Toivonen H. Sampling large databases for association rules. In Proc. 22nd VLDB, pages 134-145, Sept. 1996.

[14] Han J., Pei J., and Yin Y., Mining frequent patterns without candidate generation. In SIGMOD'00, pages 1–12, 2000.

[15] Thakur R. S., Jain R. C., Pardasani K. R., Graph Theoretic Based Algorithm for Mining Frequent Patterns. In Proc. IEEE World Congress on Computational Intelligence, Hong kong, pages 629-633, June 2008.

[16]D.S Rajput, R.S. Thakur, G.S. Thakur "Rule Generation from Textual Data by using Graph Based Approach" Published in International Journal of Computer Application

(IJCA) ISSN: 0975 – 8887, New york USA, ISBN: 978-93-80865-11-8, Volume 31– No.9, October 2011 pp 36-43.

[17]D.S Rajput, R.S. Thakur, G.S. Thakur, "Fuzzy Association Rule Mining based Frequent Pattern Extraction from Uncertain Data" presented in IEEE 2nd World Congress on Information and Communication Technologies (WICT-2012) October 30-November 02, 2012 in IIITM Trivandrum, ISBN: 978-1-4673-4804-1 pp 709-714.

[18]Neelu Khare, Neeru Adlakha and K. R. Pardasani, "Karnaugh Map Model for Mining Association Rules in Large Databases", (IJCNS) International Journal of Computer and Network Security, Vol. 1, No. 1, pp:16-21,October 2009.

[19] R.S Thakur, R.C. Jain, K.R.Pardasani, "Fast Algorithm for Mining Multilevel Association Rule Mining," Journal of Computer Science, Vol. 1, pp: 76-81, 2007.

[20]Neelu Khare, Neeru Adlakha, K. R. Pardasani,"An Algorithm for Mining Multidimensional Fuzzy Association Rules," International Journal of Computer Science and Information Security, Vol.5, pp.72-76, 2009.

[21]Pratima Gautam, K. R. Pardasani, "Algorithm for efficient multilevel association rule mining", Internal Journal on Computer Science and Enginnering, Vol. 02, N0.05, pp:1700-1704, 2010.

[22]R. Vijaya Prakash, Dr. Govardhan, Dr. S.S.V.N. Sarma, "Mining Frequent Itemsets from Large Data Sets using Genetic Algorithms", IJCA Special Issue on "Artificial Intelligence Techniques - Novel Approaches & Practical Applications", pp:38-43, 2011.

[23]Wakabi-Waiswa P.P., Baryamureeba V and Sarukesi K, "Generalized Association Rule Mining Using Genetic Algorithms", International Journal of Computing and ICT Research, Vol. 2 No. 1, pp:59-69, 2008.

[24]Minh Nghia Le and Yew Soon Ong, "A Frequent Pattern Mining Algorithm for Understanding Genetic Algorithms". In proceeding of: Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, 4th International Conference on Intelligent Computing, ICIC 2008, Shanghai, China, September 15-18, 2008.

[25] G. Vijay Bhasker K. Chandra Shekar V. Lakshmi Chaitanya, "Mining Frequent Itemsets for Non Binary Data Set Using Genetic Algorithm", INTERNATIONAL JOURNAL OF ADVANCED ENGINEERING SCIENCES AND TECHNOLOGIES, pp:143 – 152, 11(1), 2011.

[26]Xiaowei Yan , Chengqi Zhang , Shichao Zhang, "Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support", pp:3066–3076, 36, 2009.

[27] Pei M., Goodman E.D. Punch F. (2000) Feature extraction using Genetic Algorithm, Case Center for Computer Aide Engineering and Manufacturing W. Department of Computer Science.

[28] Goldberg, D.E: Genetic algorithms in search, optimization and machine learning Addison-Wesley (1989).

[29]http://www.obitko.com/tutorials/geneticalgorithms/crossover-mutation.php.

Author Biography

Vikrant Sabnis did his Masters from S.O.S. Computer Science, Jiwaji University, Gwalior (M.P.) in 1999.He joined Maulana Azad National Institute of Technology (MANIT), Bhopal as a Visiting Faculty from July 2000 to June 2003 in Department of Computer Applications and later enrolled as a full time Research Scholar in Computer Applications Department (MANIT), Bhopal in the year 2008 to till date. He has attended many workshops and conference of National repute.

Dr. Ramjeevan Singh Thakur is Associate Professor in the Department of Computer Applications at Maulana Azad National Institute of Technology, Bhopal, India. He is a Teacher, Researcher and Consultant in the field of Computer Science and Information Technology. He earned his Master Degree from Samrat Ashok Technology Institute, Vidisha (M.P.) in 1999. And Ph.D. Degree (Computer Science) From Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.) in 2008.
He had a long carrier in teaching and research, including Three Year Teaching in the Department of Computer Applications at National Institute of Technology, Tiruchirapalli, Tamilnadu, India. At Present he is guiding several Ph.D. Research Scholars and handling Government Research Projects of about Rs. One Crore. He has published more than 75 Research Paper in National, International, Journals and Conferences. He has visited several Universities in USA, Hong Kong, Iran, Thiland, Malaysia, and Singapore,
His areas of interest include Data Mining, Data Warehousing, Web Mining, Text Mining, and Natural Language Processing. He has also received DST Young Scientist Award-2011 in Engineering under Fast Track Scheme, Department of Science & Technology, New Delhi, India. He is a member of the CSI, IAENG, ISTE, GAMS and IACSIT.