Identifying Best Association Rules and Their Optimization using Genetic Algorithm

Arvind Jaiswal, Gaurav Dubey

Abstract- Data mining is the analysis step of the "Knowledge Discovery in Databases", It is the process that results in the detection of new patterns in large data sets. The main aim of data mining is to pull out knowledge from an existing dataset and transform it into a flexible structure. In data mining association rule is a popular and easy method to find frequent itemsets from large datasets. In general frequent itemsets are generated from large data sets by applying association rule mining take too much computer time to compute all the frequent itemsets. By using Genetic Algorithm (GA) we can improve the results of association rule mining. Our main purpose is by Genetic Algorithm to generate high quality Association Rules, by which we can get four data qualities like accuracy, comprehensibility, interestingness and completeness. Genetic Algorithms are powerful and widely applicable stochastic search and optimization methods based on the concepts of natural selection and natural evaluation. The advantage of using genetic algorithm is to discover high level prediction rules is that they perform a global search and cope better with attribute interaction than the greedy rule induction algorithm often used in data mining. The main aim of this paper is to find all the frequent itemsets from given data sets using genetic algorithm. In this paper we are using the large dataset and our Experimental results on this dataset show the effectiveness of our approach. This paper provides the major improvement in the approaches for association rule mining using genetic algorithms.

Keywords - Genetic Algorithm (GA), Association Rule, Frequent itemset, Support, Confidence, Data Mining.

I. INTRODUCTION

Large amounts of data have been collected routinely in the course of day-to-day management in business, administration, banking, the delivery of social and health services, environmental protection, security and in politics. Such data is primarily used for accounting and for management of the customer base. Typically, management data sets are very large and constantly growing and contain a large number of complex features. While these data sets reflect properties of the managed subjects and relations, and are thus potentially of some use to their owner, they often have relatively low information density. One requires robust, simple and computationally efficient tools to extract information from such data sets. The development and understanding of such tools is the core business of data mining.

Data mining techniques operate on structured data such as corporate databases; this has been an active area of research for many years. The main tasks of Data mining are generally divided in two categories: “Predictive and Descriptive”. The objective of the predictive tasks is to predict the value of a particular attribute based on the values of other attributes, while for the descriptive ones, is to extract previously unknown and useful information such as patterns, associations, changes, anomalies and significant structures, from large databases. There are several techniques satisfying these objectives of data mining. Mining Associations is one of the techniques involved in the process mentioned above and among the data mining problems it might be the most studied ones. Discovering association rules is at the heart of data mining. Mining for association rules between items in large database of sales transactions has been recognized as an important area of database research. These rules can be effectively used to uncover unknown relationships, producing results that can provide a basis for forecasting and decision making. The original problem addressed by association rule mining was to find a correlation among sales of different products from the analysis of a large set of data. Genetic Algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators to these structures as to preserve critical information. Genetic algorithms are often viewed as function optimizer, although the ranges of problems to which genetic algorithms have been applied are quite broad. An implementation of genetic algorithm begins with a population of chromosomes. One then evaluates these structures and allocated reproductive opportunities in such a way that these chromosomes which represent a better solution to the target problem are given more chances to reproduce than those chromosomes which are poorer solutions. The goodness of a solution is typically defined with respect to the current population [1] and [3]. Genetic Algorithms are used for a number of different application areas. An example of this would be multidimensional optimization problems in which the character string of the chromosome can be used to encode the values for the different parameters being optimized [4]. The genetic algorithms are important when discovering association rules because they work with global search to discover the set of items frequency and they are less complex than other algorithms often used in data mining. The genetic algorithms for discovery of association rules have been put into practice in real problems such as commercial databases, biology and fraud detection event sequential analysis. Our main objective is to find patterns of behavior predictable outcomes and turn these into profitable opportunities.

Manuscript received on May, 2013.

Arvind Jaiswal, M.Tech (CSE), Amity School of Engineering and Technologu, Amity University, NOIDA, Uttar Pradesn, India.

Gaurav Dubey, Asst. Prof, Amity School of Computer Science, Amity University, NOIDA, Uttar Pradesn, India.

In this paper we are using a large database of toy stock management company, first we are finding the frequent item set of toys and after that we are applying genetic algorithm on these frequent item set. This genetic algorithm approach will show that maximum frequent itemsets and minimum frequent itemsets, by which so many important decisions can be taken.

II. ASSOCIATION RULE MINING (ARM)

Association Rule Mining aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories [8]. The major aim of ARM is to find the set of all subsets of items or attributes that frequently occur in many database records or transactions, and additionally, to extract rules on how a subset of items influences the presence of another subset. ARM algorithms discover high-level prediction rules in the form: IF the conditions of the values of the predicting attributes are true, THEN predict values for some goal attributes. In association rule mining there are so many methods. These methods work basically in two phases: the frequent itemset generation and the rule generation. Since the first phase is the most time consuming, all the association rule algorithms focus on the second phase. A set of attributes is termed as frequent set if the occurrence of the set within the dataset is more than a user specified threshold called minimum support. After discovering the frequent itemsets, in the second phase rules are generated with the help of another user parameter called minimum confidence.

The task of mining association rules over market basket data is considered a core knowledge discovery activity. Association rule mining provides a useful mechanism for discovering correlations among items belonging to customer transactions in a market basket database.

In general, the association rule is an expression of the form X=>Y, where X is antecedent and Y is consequent. Association rule shows how many times Y has occurred if X has already occurred depending on the support and confidence value.

Support: It is the probability of item or item sets in the given transactional data base:

support(X) = n(X) / n where n is the total number of transactions in the database and n(X) is the number of transactions that contains the item set X.Therefore, support (X=>Y) = support(XUY).

Confidence: It is conditional probability, for an association rule X=>Y and defined as

confidence(X=>Y) = support(XUY) / support(X).

Frequent itemset: Let A be a set of items, T be the transaction database and σ be the userspecified minimum support. An itemset X in A (i.e., X is a subset of A) is said to be a *frequent itemset* in T with respect to σ, if support(X)$_T$ >= σ.

Mining association rules can be broken down into the following two sub-problems:

1. Generating all itemsets that have support greater than, or equal to, the user specified minimal support. That is, generating all large itemsets.
2. Generating all the rules that have minimum confidence.

We can generate the association rule with more than one number of consequent items is generated by the following method:

1. Find the rule in which number of consequents =1.
2. For the given rules p(x _->y) and p(x-> z), the rule p(x ->yz) is generated by the intersection of both the association rules and get a new rule

p(x ->yz) =p (xyz)/p(x).

III. RELATED WORKS

A. Association Rule Mining

Let I = {I1, I2……, Im} be a set of m distinct attributes, also called literals. Ai = r is an item, where r is a domain value is attribute, Ai in a relation, R (A1, …, An). A is an itemset if it is a subset of I. D =. { ti , ti+1 ,…………, tn) is a set of transactions, called the transaction (tid, t-itemset). A transaction t contains an itemset A if and only if, for all items iÎA, i is in t-itemset. An itemset A in a transaction database D has a support, denoted as Supp (A) (we also use p(A) to stand for Supp (A)), that is the ratio of transactions in D contain A. Supp(A) = |A(t)| / |D|, Where A(t) = {t in D/t contains A}. An itemset A in a transaction database D is called a large (frequent) itemset if its support is equal to, or greater than, a threshold of minimal support (minsupp), which is given by users or experts. An association rule is an expression of the form IF A THEN C(or A -> C), A ÇC = f, where A and C are sets of items. The meaning of this expression is that transactions of the databases, which contain A, tend to contain C. Each association rule has two quality measurements: support and confidence, defined as:

1) The support of a rule A->C is the support of A ÈC, where A ÈC means both A and C occur at the same time.

2) The confidence or predictive accuracy [5] of a rule A->C is conf (A->C) as the ratio: |(A ÈC)(t)| / |A(t) or Supp(A ÈC) / Supp(A).

That is, support = frequencies of occurring patterns; confidence = strength of implication. Support-confidence framework (Agrawal et al. 1993): Let I be the set of items in database D, A, C ÍI be itemset, A ÇC = f, p(A) is not zero and p(C) is not zero. Minimal support minsupp) and minimal confidence (minconf) are given by users or exparts. Then A->C is a valid rule if

1. Supp (AÈC) is greater or equal to minsupp,
2. Conf (A->C) is greater or equal to minconf.

IV. GENETIC ALGORITHM

Genetic Algorithm (GA) is an artificial intelligence procedure. It is based on the theory of natural selection and evolution. a genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

GAs are one of the best ways to solve a problem for which little is known. They are a very general algorithm and so will work well in any search space. The Genetic Algorithm [6] was developed by John Holland in 1970. GA is stochastic search algorithm modeled on the process of natural selection, which underlines biological evolution [7].

GA has been successfully applied in many search, optimization, and machine learning problems.

GA works in an iterative manner by generating new populations of strings from old ones. Every string is the encoded binary, real etc., version of a candidate solution. An evaluation function associates a fitness measure to every string indicating its fitness for the problem [9].

Chromosome: A chromosome (also sometimes called a genome) is a set of parameters which define a proposed solution to the problem that the genetic algorithm is trying to solve. The chromosome is often represented as a simple string; although a wide variety of other data structures are also used. We have to redefine the Chromosome representation for each particular problem, along with its fitness, mutate and reproduce methods.

Gene: A Gene is a part of chromosome. A gene contains a part of solution. For example if 162759 is a chromosome then 1, 6, 2, 7, 5 and 9 are its genes.

Fitness: Fitness (often denoted ω in population genetics models) is a central idea in evolutionary theory. It can be defined either with respect to a genotype or to a phenotype in a given environment. In either case, it describes the ability to both survive and reproduce, and is equal to the average contribution to the gene pool of the next generation that is made by an average individual of the specified genotype or phenotype. If differences between alleles at a given gene affect fitness, then the frequencies of the alleles will change over generations; the alleles with higher fitness become more common [1] and [10].

Standard GA apply genetic operators such *selection*, *crossover* and *mutation* on an initially random population in order to compute a whole generation of new strings. GA runs to generate solutions for successive generations. The probability of an individual reproducing is proportional to the goodness of the solution it represents. Hence the quality of the solutions in successive generations improves. The process is terminated when an acceptable or optimum solution is found. GA is appropriate for problems which require optimization, with respect to some computable criterion. The functions of genetic operators are as follows:

1) *Selection:* Selection deals with the probabilistic survival of the fittest, in that, more fit chromosomes are chosen to survive. Where fitness is a comparable measure of how well a chromosome solves the problem at hand.

2) *Crossover:* This operation is performed by selecting a random gene along the length of the chromosomes and swapping all the genes after that point.

3) *Mutation:* Alters the new solutions so as to add stochasticity in the search for better solutions. This is the chance that a bit within a chromosome will be flipped (0 becomes 1, 1 becomes 0).

Essentially, Genetic algorithms are a method of "breeding" computer programs and solutions to optimization or search problems by means of simulated evolution. Processes loosely based on natural selection, crossover, and mutation are repeatedly applied to a population of binary strings which represent potential solutions.

Genetic algorithm was first developed by John Holland in 1975. It is based on the idea of survival of the fittest and the greedy approach and performs very well global search with less time. The GA works as follows:

1. An initial population is created. A Population is a group of individuals (Chromosomes) and represents a candidate solution. A Chromosome is a string of genes.
2. Select chromosomes with higher fitness.
3. Crossover between the selected chromosomes to produce new offspring with better higher fitness
4. Mutate the new chromosomes if needed.
5. Terminate when an optimum solution is found.

This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

- A solution is found that satisfies minimum criteria
- Fixed number of generations reached
- Allocated budget (computation time/money) reached
- The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
- Manual inspection
- Combinations of the above

Ghosh et al. [11] proposed an algorithm to extract frequent itemsets using genetic algorithms.

V. METHODOLOGY

In this paper the Genetic algorithm (GA) is applied on large data sets to discover the frequent itemsets. We first load the sample of records from the transaction database that fits into memory.

The genetic learning starts as follows. An initial population is created consisting of randomly generated transactions. Each transaction can be represented by a string of bits. Our proposed genetic algorithm based method for finding frequent itemsets repeatedly transforms the population by executing the following steps:

(1) *Fitness Evaluation:* The fitness (i.e., an objective function) is calculated for each individual.

(2) *Selection:* Individuals are chosen from the current population as parents to be involved in recombination.

(3)*Recombination:* New individuals (called offspring) are produced from the parents by applying genetic operators such as crossover and mutation.

(4) *Replacement:* Some of the offspring are replaced with some individuals (usually with their parents).

One cycle of transforming a population is called a generation. In each generation, a fraction of the population is replaced with offspring and its proportion to the entire population is called the generation gap (between 0 and 1).

We can formally state the task of mining association rules over market basket as follows: let P={P1, P2, …, Pn} be the set of products and T={T1, T2, …, Tn} be the set of transactions in the database. Each of the transaction Ti has a unique ID and contains a subset of the items in I, called itemset. An association rule is an implication among itemsets of the form, X->Y, where X U Y is subset of I and X∩Y = Ø [1][2].

An itemset can be a single item (e.g. mineral water) or a setof items (e.g. sugar, milk, red tea). The quality of the association rules can be measured by using two important basic measures, support(S) and confidence(C) [13]. Support(S) of an association rule is the percentage of transactions in the database that contain the itemset X U Y. Confidence (C) of an association rule is the percentage/fraction of the number of transactions that contain X U Y to the total number of records that contain X. Confidence factor of X->Y can be defined as:

$$c(x \rightarrow y) = S(x \cup y) / S(x) \qquad (1)$$

Most of the association rule algorithms generate the frequent itemsets, itemsets that are greater than a minimum support and then generate association rules that have confidence greater than minimum confidence. However, more metrics such as Comprehensibility and Interestingness can be used to have more interesting association rules [12].

Comprehensibility is measured by the number of attributes involved in IF part (condition) of the rule with respect to the THEN (prediction) part of the rule because the rule is more comprehensible if the conditions is less than the prediction. Comprehensibility of an association rule (X->Y) is measured as:

$$C(x \rightarrow y) = \log(1+|y|) + \log(1+|x \cup y|) \quad (2)$$

where |Y| and |XUY| are the number of attributes in the consequent side and the total rule, respectively. Interestingness measures how much interesting the rule is[10]. The interestingness of an association rule (X->Y) is measured as:

$$Inter(x \rightarrow y) = [S(x \cup y) / S(x)] * [S(x \cup y) / S(y)$$
$$* [1 - S(x \cup y) / |d|]]$$

Where |D| is the total number of records/transactions in the database. Using several measures, the association rule problem can be considered as a multi-objective problem.

We have the database of toys stock transaction, and we are using 5 types of product to find frequent itemsets.

Products	Abbreviation
Electronic Toys	E
Inflatable toys	I
Soft toys	S
Puzzel games	P
Video games	V

Table1 (a)

Transactions	Products
1	EIPV
2	ISV
3	EIPV
4	EISV
5	EISPV
6	ISP

Table1 (b)

Table1. (a) products/items database (b) Database transactions

Table. 1(a) an d (b) depict a small grocery sales database consisting of five products P = {E, I, S, P, V} and six transactions table that illustrates the purchase of items by customers.

Itemsets	Support
I	100%
V, IV	83%
E, S, P, EI, IS, IP, EIV	67%
EP, SV, PV, EIP, EPV, ISV, IPV, EIPV	50%

Table2 (a)

Association rules	Support	Confidence	Comprehensibility	Interest-ingness
E->I	67%	1	2.58	.59
EI->P	50%	.75	3	.51
V->IS	50%	.75	3	.51
I->V	83%	.83	2.58	.71

Table2 (b)

Table. 2(a) shows all frequent itemsets containing at least three products and minimum support 50%. Table. 2(b) illustrates sample association rules with four metrics (Support, Confidence, Comprehensibility and Interestingness).

In this paper, we choose to deal with the association rule mining as a multi-objective problem rather than one objective problem and to adopt the multi-objective evolutionary algorithm for mining association rules [14], [15] with emphasize on genetic algorithms. Genetic algorithm is an iterative procedure that is appropriate for situations such as large and complex search space and optimization problems. In order to use the genetic algorithm, the following points must be addressed : Encoding/decoding schemes of chromosomes, Population size, Fitness value, Selection, Crossover and Mutation.

In this paper the genetic algorithm is applied over the rules fetched from Apriori association rule mining. The proposed method for generating association rule by genetic algorithm is as follows:

1. Start
2. Load model of records from the database that fits in the memory.
3. Apply Apriori algorithm to find the frequent itemsets with the minimum support. Suppose A is set of the frequent item set generated by Apriori algorithm.
4. Set B= Ø where B is the output set, which contains the association rule.
5. Input the termination condition of genetic algorithm.
6. Represent each frequent item set of A as binary string using the combination of representation specified in method 2 above.
7. Select the two members from the frequent item set using Roulette Wheel sampling method.

8. Apply the crossover and mutation on the selected members to generate the association rules.
9. Find the fitness function for each rule x->y and check the following condition.
10. if (fitness function > min confidence)
11. set B = B U {x ->y}
12. If the desired number of generations is not completed, then go to Step 3.
13. Stop

If a data set contains p items, the total number of association rule is 3p-2(p+1)+1 (without considering the negation of item). The total no. of rule generated for a database with n item is

(2p-2)*2p+(2(p-1)-2)*2(p-1)+………

In general we can say that number of rule generated including the negation of itemset is

$$\sum_{i=2}^{p} (2^{i} -2)*(2^{i})$$

VI. RESULTS

In this paper we are using the database from toy stock database (Mayur Packing company, mumbai) to show the effectiveness of the proposed algorithm. This database contains the following four attributes namely,

1. Age of the children with three values:

(i) Kids(0-5) (ii) Master child (6-12) (iii) teenagers(13-18)

2. Type of toys:

(i) Normal (ii) Electronic

3. Normal Toys:

(i) soft (ii) inflatable (iii) Puzzel

4. Electronic toys:

(i) Video game (ii) Movable toys

For each attribute we need log2m bits, where m is the number of attribute values. Let us represent the member of data base in the binary string with the hybrid combination of above representation.

For example, the binary string 01 **00** 1 **01** 0 **11** 1 **11** in which bold letter represent the presence (11), absence (00), not consider (01).

The columns in the database are subjects and their numerical value depends on attributes, As described earlier, the implementation of GAs is applied to the rules obtained by applying Apriori association rule mining on the database. The database is chosen randomly. The crossover and mutation probabilities were taken respectively as 0.1 and 0.005.

The frequent item set generated for the given database in Table 3 is as follows.

Table 3

(x=1),(x=2),(x=3),(y=1),(y=2),(z=1),(z=2),(u=1),(u=2)
(x=1,y=2),(x=1,z=1),(x=1,u=2),(x=2,y=1),(x=2,y=2), (x=2,z=1)
(x=2,z=2),(x=3,y=1),(x=3,z=1),(x=3,u=1),(y=1,z=1),(y=1,u=1)
(y=2,z=1),(y=2,z=2),(z=1,u=1),(z=1,u=2),((z=2,u=1)
(x=1,y=2,z=1),(x=1,y=2,u=2),(x=1,z=1,u=1),(x=2,y=1,u=1)

Here x, y, z, u represent the age of children, toy types, normal toys , and electronic toys respectively. The numerical value represents the attribute value of corresponding attribute. Table 3 shows the rules evolved from generated frequent itemsets with minimum support >=20% and minimum confidence>=60%.

VII. CONCLUSION

A great deal of association rules and frequent itemsets research has been published over the past 15-20 years. For the most part, however, researchers tend to focus on Apriori for large dataset. Given the size of the underlying market-basket databases, coupled with the availability of modestly priced hardware and the advantages of genetic algorithm. We have dealt with a challenging association rule mining problem of finding frequent itemsets using our proposed GA based method. The frequent itemsets are generated using the Apriori association rule mining algorithm. The genetic algorithm has been applied on the generated frequent itemsets to generate the rules containing positive attributes, the negation of the attributes with the consequent part consists of single attribute and more than one attribute.The method, described here is very simple and efficient one. This is successfully tested for different large data sets. The results reported in this paper are correct and appropriate. However, a more extensive empirical evaluation of the proposed method will be the objective of our future research.

REFERENCES

1. Rupali Haldulakar and Prof. Jitendra Agrawal, “Optimization of Association Rule Mining through Genetic Algorithm”, International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 3 Mar 2011, pp. 1252-1259.
2. Manish Saggar, Ashish Kumar Agarwal and Abhimunya Lad, “Optimization of Association Rule Mining using Improved Genetic Algorithms”IEEE 2004.
3. Anandhavalli M, Suraj Kumar Sudhanshu, Ayush Kumar and Ghose M.K., “Optimized association rule mining using genetic algorithm”, Advances in Information Mining, ISSN: 0975–3265, Volume 1, Issue 2, 2009, pp-01-04.
4. Farah Hanna AL-Zawaidah, Yosef Hasan Jbara and Marwan AL-Abed Abu-Zanona, “An Improved Algorithm for Mining Association Rules in Large Databases”, World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741 Vol. 1, No. 7, 2011, pp. 311-316.
5. L.Jain, A.Abraham, R.Goldberg, “Evolutionary Multiobjective Optimization”, Second Edition, Springer 2005).International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.2, No.1, January 2012.
6. Pei M., Goodman E.D., Punch F. (2000) *Feature Extraction using genetic algorithm, Case Center for Computer-Aided Engineering and Manufacturing W. Department of Computer Science.*
7. Stuart J. Russell, Peter Norvig (2008) *Artificial Intelligence: A Modern Approach*.
8. Han J., Kamber M. *Data Mining*: *Concepts & Techniques*, Morgan & Kaufmann, 2000.
9. Goldberg, David E. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*. Addison Wesley. pp. 41.
10. 1S. Dehuri,A. K. Jagadev, A. Ghosh and R. Mall, “Multi-objective Genetic Algorithm for Association Rule Mining Using a Homogeneous Dedicated Cluster of Workstations”, American Journal of Applied Sciences 3 (11), 2006, pp. 2086-2095.
11. S. Ghosh., S. Biswas., D. Sarkar., and P. Sarkar, “Mining Frequent Itemsets Using Genetic Algorithm,” international Journalof Artificial Intelligence & Applications (IJAIA), Vol.1, No.4, October 2010
12. A. Ghosh and B. Nath, “Multi−objective rule mining using genetic algorithms,” Information Sciences 163, pp 123-133, 2004.

13. S. Kotsiantis and D. Kanellopoulos, “Association Rules Mining: A Recent Overview,” GETS International Transactions on Computer Science and Engineering, Vol.32(1), 2006, pp.71-82.
14. M. Fonesca and J. Fleming, “Multi-objective Optimization and Multiple Constraint Handling with Evolutionary Algorithms,” Part I: A Unified Formulation. IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans, 28(1), pp. 26-37, 1998.
15. A. Freitas, “Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery,” Advances in evolutionary computing: theory and applications, pp 819 – 845, 2003.
16. LIU, B., HSU, W., CHEN, S., AND MA, Y. 2000. *Analyzing the Subjective Interestingness of Association Rules*. *IEEE Intelligent Systems*.
17. David Beasley et al., “An overview of genetic algorithms”, Part 1 & 2, University Computing, Vol. 15, No. 2 & 4, pp.58-69 & 170-181, 1993.
18. GHOSH, A., NATH, B. 2004. *Multi-objective rule mining using genetic algorithms*. Information Sciences 163 pp 123-133.

AUTHORS PROFILE

Mr. Arvind Jaiswal is student of M.Tech (CSE) in Amity School of Engineering and Technology, Amity University, NOIDA(UP). He has published 11 research papers in reputed International Journal, International and National conferences.

Mr. Gaurav Dubey is working as Assistant Professor in Amity School of Computer Science, Amity University, NOIDA (UP).He has 13 years of Teaching Experience. He has published 13 research papers in reputed International journals, International conferences and springer Journal. His area of expertise is Data Mining and Date Warehousing.