Volume 6, Issue 2, February 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Enhanced Association Rule Mining in Vertical Distributed Databases using Secure Multi Party Algorithm

Kranthi Kumar K*
Asst. Professor, Department of CSE
SNIST, Hyderabad, India

Sunil Bhutada
Assoc. Professor, Department of IT
SNIST, Hyderabad, India

Pavani Arukonda
PG Scholar, Department of IT
SNIST, Hyderabad, India

Abstract— Association rule mining is an active data mining research area and most algorithms cater to a distributed environment. This paper addresses problem of association rule mining where transactions are distributed across the sources. In this paper, we propose a protocol for secure mining of association rules in vertically distributed databases which is based on FDM (Fast Distributed Mining) algorithm and Secure Multi Party algorithm on vertically Distributed Databases. Here transactional attributes are distributed across the databases and goal is to find all association rules globally without revealing the transaction data. Our protocol is more efficient for discovering frequent item sets with minimum support levels. In addition, it offers enhanced privacy with respect to the protocol used previously.

Keywords— Association rules, Distributed Data bases, Multi party computation, Privacy.

I. INTRODUCTION

Data mining technology has emerged as a means for identifying patterns and trends from large quantities of data. Mining encompasses various algorithms such as clustering, classification, association rule mining and sequence detection. Traditionally, all these algorithms have been developed within a centralized model, with all data being gathered into a central site, and algorithms being run against that data.

Association rule mining finds all rules in the databases that satisfy some minimum support and minimum confidence constraints. Many algorithms are used to enhance the privacy and security of data. By vertically partitioned, we mean that each site contains some elements of a transaction. Using the traditional "market basket" example, one site may contain grocery purchases, while another has clothing purchases. Using a key such as credit card number and date, we can join these to identify relationships between purchases of clothing and groceries. However, this discloses the individual purchases at each site, possibly violating consumer privacy agreements.

There are more realistic examples. In the sub-assembly manufacturing process, different manufacturers provide components of the finished product. Cars incorporate several subcomponents; tires, electrical equipment, etc.; made by independent producers. Again, we have proprietary data collected by several parties, with a single key joining all the data sets, where mining would help detect/predict malfunctions. The recent trouble between Ford Motor and Firestone Tire provide a real-life example. Ford Explorers with Firestone tires from a specific factory had tread separation problems in certain situations, resulting in 800 injuries. Since the tires did not have problems on other vehicles, and other tires on Ford Explorers did not pose a problem, neither side felt responsible. The delay in identifying the real problem led to a public relations nightmare and the eventual replacement of 14.1 million tires[1]. Many of these were probably fine – Ford Explorers accounted for only 6.5 million of the replaced tires [2]. Both manufacturers had their own data – early generation of association rules based on all of the data may have enabled Ford and Firestone to resolve the safety problem before it became a public relations nightmare.

Informally, the problem is to mine association rules across two databases, where the columns in the table are at different sites, splitting each row. One database is designated the primary, and is the initiator of the protocol. The other database is the responder. There is a join key present in both databases. The remaining attributes are present in one database or the other, but not both. The goal is to find association rules involving attributes other than the join key.

The protocol that we propose here computes a parameterized family of functions, which we call threshold functions, in which the two extreme cases correspond to the problems of computing the union and intersection of private subsets .Those are in fact general-purpose protocols that can be used in other contexts as well. Another problem of secure multiparty computation that we solve here as part of our discussion is the set inclusion problem; namely, the problem where Alice holds a private subset of some ground set, and Bob holds an element in the ground set, and they wish to determine whether Bob's element is within Alice's subset, without revealing to either of them information about the other party's input beyond the above described inclusion. The proposed protocol improves upon that in [3] in terms of simplicity and efficiency as well as privacy. In particular, our protocol does not depend on commutative encryption and oblivious transfer (what simplifies it significantly and contributes towards much reduced communication and computational costs). While our solution is still not perfectly secure, it leaks excess information only to a small number

(three) of possible coalitions, unlike the protocol of [3] that discloses information also to some single players, that our protocol may leak is less sensitive than the excess information leaked by the protocol of [3].

II. RELATED WORK

The centralized data mining model assumes that all the data required by any data mining algorithm is either available at or can be sent to a central site. A simple approach to data mining over multiple sources that will not share data is to run existing data mining tools at each site independently and combine the results [4, 5, 6]. However, this will often fail to give globally valid results. Issues that cause a disparity between local and global results include:

- Values for a single entity may be split across sources. Data mining at individual sites will be unable to detect cross-site correlations.
- The same item may be duplicated at different sites and will be over-weighted in the results.
- Data at a single site is likely to be from a homogeneous population, hiding geographic or demographic distinctions between that population and others.

Algorithms have been proposed for distributed data mining. Cheung et al. proposed a method for horizontally partitioned data[7], and more recent work has addressed privacy in this model[8]. Distributed classification has also been addressed. A meta-learning approach has been developed that uses classifiers trained at different sites to develop a global classifier [5, 6]. This could protect the individual entities , but it remains to be shown that the individual classifiers do not disclose private information. Recent work has addressed classification using Bayesian Networks in vertically partitioned data [9], and situations where the distribution is itself interesting with respect to what is learned [10]. However, none of this work addresses privacy concerns.

There has been work in cooperative computation between entities that mutually distrust one another. Secure two party computation was first investigated by Yao [11], and later generalized to multiparty computation. The seminal paper by Goldreich proves existence of a secure solution for any functionality[12]. The idea is as follows: the function F to be computed is first represented as a combinatorial circuit, and then the parties run a short protocol to securely compute every gate in the circuit. Every participant gets corresponding shares of the input wires and the output wires for every gate. This approach, though appealing in its generality and simplicity, means that the size of the protocol depends on the size of the circuit, which depends on the size of the input. This is inefficient for large inputs, as in data mining. In [13], relationships have been drawn between several problems in Data Mining and Secure Multiparty Computation. Although this shows that secure solutions exist, achieving efficient secure solutions for privacy preserving distributed data mining is still open.

Previous work in privacy preserving data mining has considered two related settings. One, in which the data owner and the data miner are two different entities, and another, in which the data is distributed among several parties who aim to jointly perform data mining on the unified corpus of data that they hold. In the first setting, the goal is to protect the data records from the data miner. Hence, the data owner aims at anonymizing the data prior to its release. The main approach in this context is to apply data perturbation [14], [2]. The idea is that the perturbed data can be used to infer general trends in the data, without revealing original record information.

In the second setting, the goal is to perform data mining while protecting the data records of each of the data owners from the other data owners. This is a problem of secure multiparty computation. The usual approach here is cryptographic rather than probabilistic. Lindell and Pinkas [15] showed how to securely build an $ID3 decision\ tree$ when the training set is distributed horizontally. Lin et al. [16] discussed secure clustering using the $EM\ algorithm$ over horizontally distributed data. The problem of distributed association rule mining was studied in [10], [17], [18] in the vertical setting, where each party holds a different set of attributes, and in [3] in the horizontal setting. Also the work of [19] considered this problem in the horizontal setting, but they considered large-scale systems in which, on top of the parties that hold the data records (resources) there are also managers which are computers that assist the resources to decrypt messages; another assumption made in [19] that distinguishes it from[3] and the present study is that no collusions occur between the different network nodes — resources or managers.

The problem of secure multiparty computation of the union of private sets was studied in [9], [8], [11], as well as in [3]. Freedman et al. [8] present a $privacy - preserving\ protocol$ for set intersections. It may be used to compute also set unions through set complements, since $A \cup B = A \cap B$. Kissner and Song [11] present a method for representing sets as polynomials, and give several privacy-preserving protocols for set operations using these representations. They consider the threshold set union problem, which is closely related to the $threshold\ function$. The communication overhead of the solutions in those two works, as well as in [3]'s and in our solutions, depends linearly on the size of the ground set. However, as the protocols in [8], [11] use $homomorphic\ encryption$, while that of [3] uses $commutative\ encryption$, their computational costs are significantly higher than ours. The work of Brickell and Shmatikov [9] is an exception, as their solution entails a communication overhead that is logarithmic in the size of the $ground\ set$. However, they considered only the case of two players, and the logarithmic communication overhead occurs only when the size of the intersection of the two sets is bounded by a constant. The problem of $set\ inclusion$ can be seen as a simplified version of the privacy-preserving keyword search.

III. PROPOSED SYSTEM

A. Proposed System Architecture

In the below Fig.1, depicted the proposed system architecture.

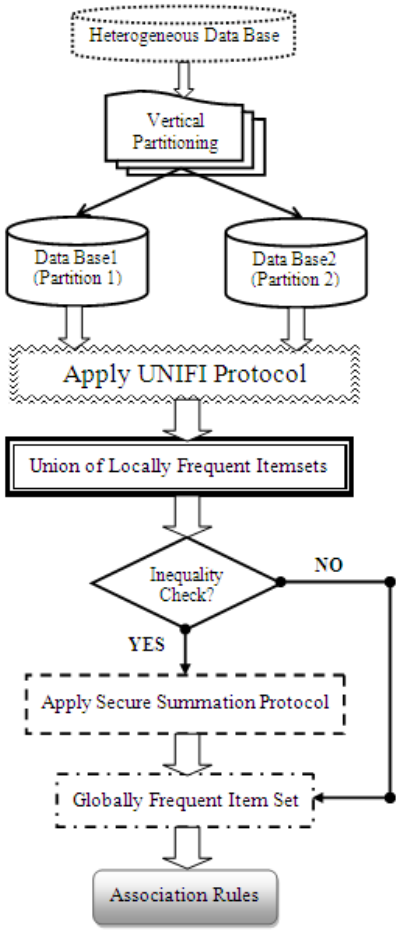

Fig.1. *Proposed System Architecture*

B. Proposed System algorithm

The Proposed algorithm proceeds as follows.

Step 1: Consider heterogeneous Data Base which contains some set of attributes.

Step 2: Perform Vertical Partitioning of the database between two parties Database 1 and Database 2.

Step 3: Find locally frequentitem sets in each individual data base.

Step 4: Apply *UNIFI Protocol* on locally frequent item sets of each data base to unify the item sets without revealing the individual information.

Step5: Identify the Globally frequent item set using Secure Summation protocol and *SETINC Protocol* . This computation must not reveal local support and size of item sets.

Step 6: Identify all Association Rules on Globally frequent Item sets.

The following paragraphs are discussed in detailabout main parts of the proposed system, which influence the finding of *association rules* for *Vertical Distributed Databases (VDD)*.

1) Identifying the Locally Frequent Item sets

To find out if a particular *item set* is frequent, we count the number of records where the values for all the attributes in the item set are 1. This translates into a simple mathematical problem, given the following definitions:

Let the total number of attributes be $l + m$, where A has l attributes, $A1$ through Al, and B has the remaining m attributes $B1$ through Bm. Transactions/records are a sequence of $l + m$, either 1^s *or* 0^s.Let k be the support threshold required, and n be the total number of transaction/records. Let $\vec{X}$ and $\vec{Y}$ represent columns in the database, i.e., x_i= 1 if row i has value 1 for attribute X. The scalar (or dot).Check it once product of two cardinality n vectors$\vec{X}$ and $\vec{Y}$.

We present an efficient way to compute scalar product $\vec{X}$*$\vec{Y}$without either side disclosing its vector. First we will show how to generalize the above protocol from two item sets to general association rules without sharing information other than through scalar product computation.

Generalizing this protocol to a $w - item\ set$ is straightforward. Assume A has p attributes $a_1, \ldots, a_p$ and B has $q$attributes$b_1, \ldots . b_q$, and we want to compute the frequency of the $w = p + q - item\ set$ $(a_1 \ldots . a_p; b_1 \ldots . b_q)$. Each item in $\vec{X}(\vec{Y})'s$ composed of the product of the corresponding individual elements,

i.e.,

$$x_i = \pi_{j=1}^{p} a_j \quad (1)$$

and

$$y_i = \pi_{j=1}^{q} b_j \quad (2)$$

This computes $\vec{X}$ and $\vec{Y}$as shown in $Eq.1$ and $Eq.2$without sharing information between $A$and B. The scalar product protocol then securely computes the frequency of the entire $w-item\ set$.

2) Unify the Locally Frequent Item sets

We denote by $F_s^{k-1}$the set of all globally frequent$(k-1)-itemsets$, and by $Ap(F_s^{k-1})$the set of $k-itemsets$ that the Apriori algorithm generates when applied on F_s^{k-1}.All players can compute the set $Ap(F_s^{k-1})$ and decide on an ordering of it(Since all item sets are subsets of $A=\{a_1,\ldots,a_L\}$, they may be viewed as binary vectors in $\{0,1\}^L$and, as such, they may be ordered lexicographically).

Then, since the sets of locally frequent $k-item\ sets$, $C_s^{k,m}, 1 \le m \le M$, are subsets of $Ap(F_s^{k-1})$, they may be encoded as binary vectors of length$n_k = |Ap(F_s^{k-1})|$.

The binary vector that encodes the union $C_s^k = \cup_{m=1}^M C_s^{k,m}$ *is* the OR of the vectors that encode the item sets $C_s^{k,m}$, $1 \le m \le M$. Hence, the players can compute the union by invoking Protocol $THRESHOLD-C$ on their binary input vectors.

3) Identifying the Globally Frequent Item sets

Protocols UNIFI-KC and UNIFI yield the set C_s^kthat consists of all item sets that are locally s-frequent in at least one site. Those are the $k-item\ sets$ that have potential to be also globally s-frequent. In order to reveal which of those item sets is globally s-frequent there is a need to securely compute the support of each of those item sets. That computation must not reveal the local support in any of the sites. Let $x$be one of the candidate item sets in C_s^k.

Then $x$is globally s-frequent if and only if

$$\Delta(x) := supp(x) - sN = \Sigma_{m=1}^M (supp_m(x) - sN_m) \ge 0 \quad (3)$$

We describe here the solution that was proposed by Kantarcioglu and Clifton. They considered two possible settings. If the required output includes all globally s-frequent itemsets, as well as the sizes of their supports, then the values of $\Delta(x)$ can be revealed for all $x \in C_s^k$.

In such a case, those values may be computed using a secure summation protocol (e.g. [5]), where the private addend of $P_m$is $supp_m(x)$-sN_mThe more interesting setting, however, is the one where the support sizes are not part of the equired output. We proceed to discuss it.

As $|\Delta(x)| \le N$, an item set $x \in C_s^k$is s-frequent if and only if $\Delta(x)$ mod $q \le N$,for $q = 2N+1$. The idea is to verify that inequality by starting an implementation of the secure summation protocol of [5] on the private inputs$\Delta_m(x)$= $supp_m(x)-sN_m$, modulo q. In that protocol, all players jointly compute random additive shares of the required sum $\Delta(x)$ and then, by sending all shares to, say,P_1, he may add them and reveal the sum. If, however, $P_M$withholds his share of the sum, then P_1will have one random share, $s_1(x)$, of $\Delta(x)$, and $P_M$will have a corresponding share, $s_M(x)$; namely, $s_1(x)+s_M(x) = \Delta(x)$ mod q. It is then proposed that the two players execute the generic secure circuit evaluation of [32] in order to verify whether

$$(s_1(x) + s_M(x)) mod\ q \le N \quad (4)$$

Those circuit evaluations may be parallelized for all $x \in C_s^k$.We observe that inequality $Eq.4$ holds if and only if

$$s_1(x) \in \Theta(x) := \{(j - s_M(x))\ mod\ q : 0 \le j \le N\} \quad (5)$$

As $s_1(x)$ is known only to $P1$ while $\Theta(x)$ is known only to P_M, the verification of the set inclusion in Eq.5 can also be carried out by means of Protocol SETINC. However, the ground set Ω in this case is $Z_q = 2N+1$, which a large set is typically. (Recall that when Protocol SETINC is invoked from UNIFI, the ground set Ω is Z_M+1, which is usually a small set.) Hence, Protocol SETINC is not useful in this case, and, consequently, Yao's generic protocol remains, for the moment, the protocol of choice to securely verify inequality Eq.4. Yao's protocol is designed for the two-party case. In our setting, as$M > 2$, there exist additional semi-honest players. An interesting question which arises in this context is whether the existence of such additional semi-honest players may be used to verify inequalities like Eq.4, even when the modulus is large, without resorting to costly protocols such as oblivious transfer.

4) Identifying all Association Rules

Once the set $F_s$of all s-frequent Itemsets is found, we may proceed to look for all $(s,c)-association\ rules$ (rules with support at least sN and confidence at least c), as described in [3].

For $X, Y \in F_s$, where $X \cap Y = \emptyset$, the corresponding association rule $X \Rightarrow Y$has confidence at least $c$if and only if $supp(X \cup Y)/supp(X) \ge c$, or, equivalently,

$$C_{X,Y} = \Sigma_{m=1}^M (supp_m(X \cup Y) - c \cdot supp_m(X)) \ge 0 \quad (10)$$

Since $|C_{X,Y}| \le N$, then by taking $q = 2N+1$, the players can verify inequality Eq.10, in parallel, for all candidate association rules.

IV. EXPERIMENTAL EVALUATION

1. Generation of Heterogeneous Databases.

The databases that we used in our experimental evaluationare synthetic databases that were generated using the sametechniques that were introduced in [22] and then used also insubsequent studies such as [7], [3], [23]. Table 1 gives theparameter values that were used in generating the syntheticdatabase. The reader is referred to [7], [3], [23] for a descriptionof the synthetic generation method and the meaningof each of those parameters. The parameter values that weused here are similar to those used in [7], [3], [23].

Table.1. Parameters for generating Heterogeneous Database

Parameter	Interpretation	Value
N	Number of transactions in the whole	1000
L	database	500
At	Number of items	10
Af	Transaction average size	4
Nf	Average size of maximal potentially large	800
CS	Itemsets	5
PS	Number of maximal potentially large	60
Cor	Itemsets	0.5
MF	Clustering size	900
	Pool size	
	Correlation level	
	Multiplying factor	

2. Vertically Partitioning of Database

Given a generated synthetic database D of N transactions and a number of players M, we create an artificial split of D into M partial databases, $D_m, 1 \leq m \leq M$, in the following manner: For each $1 \leq m \leq M$ we draw a random number w_m from a normal distribution with mean 1 and variance 0.1, where numbers outside the interval $[0.1, 1.9]$ are ignored.Then, we normalize those numbers so that $\Sigma_{M_{m=1}} w_m = 1$.Finally, we randomly split D into m partial databases of expected sizes of $w_m N, 1 \leq m \leq M$, as follows: Each transaction $t \in D$ is assigned at random to one of the partial databases, so that $Pr(t \in D_m) = w_m, 1 \leq m \leq M$.

Table.2. Description of vertical Database

Parameter	Interpretation	Partition1	Partition2
N	Number of transactions in the whole database	500	500
L	Number of items	250	250
At	Transaction average size	5	5
Af	Average size of maximal potentially large	2	2
Nf	Itemsets	400	400
CS	Number of maximal potentially large Itemsets	5	5
PS	Clustering size	30	30
Cor	Pool size	0.5	0.5
MF	Correlation level	900	900
	Multiplying factor		

3. Experimental Setup

We compared the performance of two secure implementations of the FDM algorithm. In the first implementation(denoted $FDM - KC$), we executed the unificationstep (Step 4 in FDM) using Protocol $UNIFI - KC$, where thecommutative cipher was 1024-bit RSA [25]; in the secondimplementation (denoted FDM) we used our Protocol $UNIFI$,where the keyed-hash function was $HMAC$ [4]. In both implementations,we implemented Step 5 of the FDM algorithm inthe secure manner. We testedthe two implementations with respect to three measures:

1) Total computation time of the complete protocols ($FDMKC$ andFDM) over all players. That measure includes the Apriori computation time, and the time to identify the globally $s-$frequent itemsets,(The latter two procedures are implemented in the sameway in both Protocols $FDM - KC$ and FDM.)
2) Total computation time of the unification protocols only ($UNIFI - KC$ and$UNIFI$) over all players.
3) Total message size.

We ran three experiment sets, where each set tested thedependence of the above measures on a different parameter:

- N— the number of transactions in the unified database
- M— the number of players, and
- s — the threshold support size.

In our basic configuration, we took $N = 1000$, $M = 10$,and $s = 0.1$. In the first experiment set, we kept M and $s$fixedand tested several values of N. In the second experiment set,we kept $N$and $s$fixed and varied M. In the third set, we kept$N$and Mfixed and varied$s$.All experiments were implemented in Java(NetBeans) andwere executed on an Intel(R)

Core(TM)i7-2620M personalcomputer with a 2.7GHz CPU, 8 GB of RAM, and the 64-bitoperating system Windows 7 Professional SP1.

REFERENCES

[1] National Highway Traffic Safety Administration.Firestone tire recall. http://www.nhtsa.dot.gov/hot/Firestone/Index.html,May 2001.

[2] Ford Motor Corporation. Corporate citizenship report. http://www.ford.com/en/ourCompany/communityAndCulture/buildingRelationships/strategicIssues /firestoneTireRecall.htm, May 2001.

[3] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge and Data Engineering*, 16:1026–1037, 2004.

[4] P. Chan. An Extensible Meta-Learning Approach for Scalable and Accurate Inductive Learning. PhD thesis, Department of Computer Science, Columbia University, New York, NY, 1996. (Technical ReportCUCS-044-96).

[5] P. Chan. On the accuracy of meta-learning for scalable data mining. Journal of Intelligent Information Systems, 8:5-28, 1997.

[6] A. Prodromidis, P. Chan, and S. Stolfo. Meta-learning in distributed data mining systems: Issues and approaches, chapter 3. AAAI/MIT Press, 2000.

[7] D. W.-L. Cheung, V. Ng, A. W.-C. Fu, and Y. Fu. Efficient mining of association rules in distributed databases. Transactions on Knowledge and Data Engineering, 8(6):911-922, Dec. 1996.

[8] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. In The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'02), June 2 2002.

[9] R. Chen, K. Sivakumar, and H. Kargupta. Distributed web mining using Bayesian networks from multiple data streams. In The 2001 IEEE International Conference on Data Mining. IEEE, Nov. 29 - Dec. 2001.

[10] R. Wirth, M. Borth, and J. Hipp. When distribution is part of the semantics: A new problem class for distributed knowledge discovery. In Ubiquitous Data Mining for Mobile and Distributed Environments workshop associated with the Joint 12th EuropeanConference on Machine Learning (ECML'01) and 5th European Conference on Principles and Practice ofKnowledge Discovery in Databases (PKDD'01), Freiburg, Germany, Sept.3-7 2001.

[11] A. C. Yao. How to generate and exchange secrets. In Proceedings of the 27th IEEE Symposium on Foundations of Computer Science, pages 162-167 IEEE, 1986.

[12] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game - a completeness theorem for protocols with honest majority. In 19th ACM Symposium on the Theory of Computing, pages 218-229, 1987.

[13] W. Du and M. J. Atallah. Secure multi-party computation problems and their applications: A review and open problems. In Proceedings of the 2001New Security Paradigms Workshop, Cloudcroft, New Mexico, Sept. 11-13 2001.

[14] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *SIGMODConference*, pages 439–450, 2000.

[15] Y. Lindell and B. Pinkas. Privacy preserving data mining. In *Crypto*,pages 36–54, 2000.

[16] X. Lin, C. Clifton, and M.Y. Zhu. Privacy-preserving clustering withdistributed EM mixture modeling. *Knowl. Inf. Syst.*, 8:68–81, 2005.

[17] J. Vaidya and C. Clifton. Privacy preserving association rule mining invertically partitioned data. In *KDD*, pages 639–644, 2002.

[18] J. Zhan, S. Matwin, and L. Chang. Privacy preserving collaborativeassociation rule mining. In *Data and Applications Security*, pages 153–165, 2005.

[19] A. Schuster, R. Wolff, and B. Gilburd. Privacy-preserving associationrule mining in large-scale distributed systems. In *CCGRID*, pages 411–418, 2004

[20] R.L. Rivest, A. Shamir, and L.M. Adleman. A method for obtainingdigital signatures and public-key cryptosystems. *Commun. ACM*,21(2):120–126, 1978.

[21] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions formessage authentication. In *Crypto*, pages 1–15, 1996.

[22] R. Agrawal and R. Srikant. Fast algorithms for mining association rulesin large databases. In *VLDB*, Pages 487–499, 1994.

[23] J.S. Park, M.S. Chen, and P.S. Yu. An effective hash based algorithmfor mining association rules. In *SIGMOD Conference*, Pages175–186, 1995.

AUTHORS PROFILE

Kranthi Kumar K[1], Currently working as Asst. Professor in the department of CSE, Sreenidhi Institute of Science and Technology, Hyderabad, India,. He has 11 years of experience in teaching. Graduated in B.Tech (CSE) from JNTU Hyderabad in 2003, M.Tech (CSE) from JNTU, Anantapur, A.P. in 2007. He is pursuing Ph.D in Computer Science & Engineering from JNTUH, Hyderabad. He has 50 publications in various international /national journals and conferences. His areas of Interests are Image Processing,

Content Based Image Retrieval, Information Retrieval Systems, Database Management Systems, Distributed Databases and Computer Networks.

Sunil Bhutada[2] Graduated in B.E.(CSE) from Amravathi University in 1993. He received Masters Degree in M.Tech.(Software Engineering), from JNT University, Hyderabad, in 2006. He worked as Software Engineer thereafter and later shifted to academics in 1998. He is currently attached with Sreenidhi Institute of Science & Technology in Hyderabad as Associate Professor in IT department. Her areas of interest include Data Mining, Information Security, Information Retrieval System Presently he is pursuing Ph.D from Jawaharlal Nehru Technological University, Hyderabad, India, in the field of Data Mining.

Pavani Arukonda[3]**,** is studying M.Tech (CN&IS) in Sreenidhi Institute of Science and Technology, Yamnampet, Gatkesar, Hyderabad. She completed B.Tech in CSE in the year 2011 from JNTUHCEJ, Karimnagar. Her Areas of Interest are Image Processing, Communication systems, Object Oriented Programming and Computer Networks.

