

MINING ELEVATED SERVICE ITEMSETS ON TRANSACTIONAL RECORDSETS USING SLICING

V.Suganthi¹, J.Kalai Vani²

¹Associate Professor, Information Technology, IFET College of Engineering, Villupuram, India

²Assistant Professor, Information Technology, IFET College of Engineering, Villupuram, India

Abstract

Large transactions take very long time to access the data and the system performance will be degraded and the speed will be reduced. In order to maintain large transactions in an easy and faster way a concept called slicing is used in this paper. Slicing is a process of grouping of two data into a single data in order to reduce the space and also to reduce the time taken to produce the data. The slicing uses two concepts as generalization and bucketization. Generalization is a process where the grouped data can be viewed separately if required in order to get a clear view of the data in the database. Bucketization is the process where the data are found with the help of the age perspective. The original data and the duplicate data are separated using this process. The slicing partitions the data both horizontally and vertically. The main advantage of the slicing is that it handles the high dimensional data.

Keywords: Generalization, Bucketization, Slicing, Attribute partitioning.

1. INTRODUCTION

Data mining is the process of processing large volumes of data (usually stored in a database), searching for patterns and relationships within that data. Data mining is the exploration and analysis of large quantities of data in order to discover meaningful patterns and rules. Data stream mining has become an emerging research topic in the data mining field, and finding frequent Itemsets is an important task in data stream mining with wide applications. Recently, utility mining is receiving extensive attentions with two issues reconsidered: First, the utility (e.g., profit) of each item may be different in real applications; second, the frequent Itemsets might not produce the highest utility. A novel algorithm named GUIDE (Generation of temporal maximal Utility Itemsets from Data streams) which can find temporal maximal utility Itemsets from data streams. A novel data structure, namely, TMUI-tree (Temporal Maximal Utility Itemsets tree), is also proposed for efficiently capturing the utility of each itemset with one-time scanning.

1) GUIDE is the first one-pass utility-based algorithm for mining temporal maximal utility Itemsets from data streams, and

2) TMUI-tree is efficient and easy to maintain. The experimental results show that our approach outperforms other existing utility mining algorithms like Two-Phase algorithm under the data stream environment.

However, mining high utility itemsets from databases is not an easy task since downward closure property [1] in frequent itemset mining does not hold. In other words, pruning search space for high utility itemset mining is difficult because a superset of a low-utility itemset may be a high utility itemset. To address this problem we propose two

concepts called generalization and bucketization with slicing.

1.1 Existing System

In the existing system the data were separate and each and every process was done individually and the processing took a long time to execute the output from the database. And also the data can be collapsed and there can be a chance for data loss to occur. In existing two algorithms were used (UP-Growth and UP-Growth+) [2]. UP-Growth is where the data are added in the database and the UP-Growth+ is where the data are deleted from the database. The UP-Growth and UP-Growth+ is managed using the UP-Growth Tree.

1.2 Disadvantages of Existing System

- Long processing time and the system performance is degraded.
- The space occupied by each data is more and thus transactions time will also be more.
- There is a chance of data loss.
- Backup of the data are taken that are deleted from the database.

2. PROPOSED SYSTEM

In this paper we propose two concepts called generalization and bucketization. Slicing is used to combine two data from the tables and generate in a single table to reduce the space. The generalization is used to view the tables separately to get a clear view of data and to check them if any error has occurred. Bucketization is a process where the original data and the duplicate data are compared and the correct result is generated with the help of age perspective. These are used to combine the data of two table into a single table in order to

reduce the space and to generate the result without waiting time and without degrading the system performance. The slicing uses two concepts they are generalization and bucketization. Generalization is the process where the data separated using the slicing is viewed according to the convenience of the user. The bucketization is the process where the data can be separated according to the age perspective as shown in Fig-1.

In Slicing, each attribute is in exactly one column. An extension is the notion of overlapping slicing, which duplicates an attribute in more than one column. Slicing is quite different from marginal publication in a number of aspects. First, marginal publication can be viewed as a special case of slicing which does not have horizontal partitioning. Slicing is a promising technique for handling high-dimensional data.

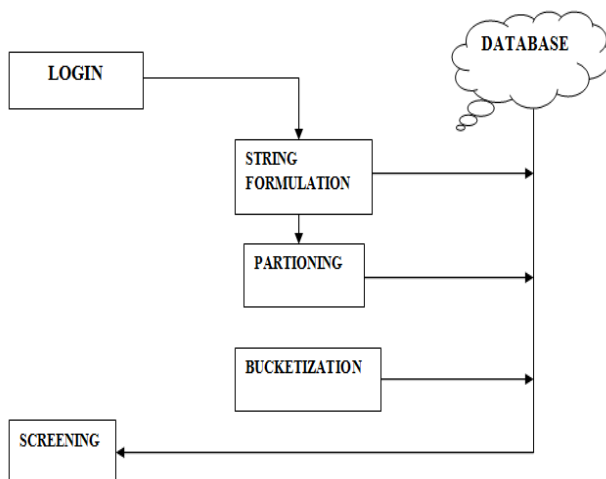


Fig -1: Overall Architecture

2.1 Advantages of Proposed Work

- As the data are combined into one table the space will be reduced.
- The transactions will be fast without degrading the systems performance.
- There is no chance of data loss.

3. DESIGN AND IMPLEMENTATION

This paper has the following modules

- Formalization of slicing
- Attribute partitioning
- Multiset
- Bucketization

3.1 Formalization of Slicing

Slicing is a promising technique for handling high dimensional data. By partitioning attributes into columns, we protect privacy by breaking the association of uncorrelated attributes and preserve data utility by

preserving the association between highly correlated attributes. Privacy-preserving data mining is the area of data mining that used to safeguard sensitive information from unsanctioned disclosure. The problem of privacy-preserving data mining has become more important in recent years because of the increasing ability to store personal data about users. A number of techniques such as randomization and k-anonymity, bucketization, generalization have been proposed in recent years in order to perform privacy-preserving data mining. For high-dimension data by using generalization significant amount of information is lost according to recent works.

3.2 Attribute Partitioning

When column generalization is applied, to achieve the same level of privacy against attribute disclosure, bucket sizes can be smaller. While column generalization may result in information loss, smaller bucket-sizes allow better data utility. Therefore, there is a trade-off between column generalization and tuple partitioning. In this paper, we mainly focus on the tuple partitioning algorithm. The trade-off between column generalization and tuple partitioning is the subject of future work. Attribute partitioning refers to the process where the tables are separated according to the tuple partitioning algorithm and the data are partitioned without any data loss.

The data are separated using slicing in order to reduce the space and to execute the result without degrading the system performance. The attributes here refers to the data that are extracted from the database after combining them into a single table using slicing. The partitioning is done based on the data that are already there in the database. Each data has its own representation in the database. Tuple partitioning is the algorithm that is used here, that is the data are partitioned according to the tuple(row) based. Thus the attributes are partitioned according to the row, to combine the data of two tables into a single table in order to reduce the space and to maintain the system performance.

3.3 Multiset

The Multiset is a process where the data are grouped and displayed so that the count of people will be given with the gender. Here the same aged people and also the people with different age are shown in different categories. By doing this the data can be easily viewed and the data can be fastly accessed. The data can be accessed very fastly so that the system performance will not be degraded.

The notion of multiset (or bag) is a generalization of the notion of set in which members are allowed to appear more than once. For example, there is a unique set that contains the elements a and b and no others, but there are many multisets with this property, such as the multiset that contains two copies of a and one of b or the multiset that contains three copies of both a and b. The Multiset is the process where the data that are combined are generated separately and viewed in order to the convenience of the user.

It also reduces the space and time to search to get the data from the database and to get a clear view of the data in the database. Large database transactions can degrade the system performance, so that the data are combined using slicing. The combined data are viewed using the Multiset generalization process.

3.4 Bucketization

Bucketization is to consider the number of matching buckets for original tuples and that for fake tuples. If they are similar enough, membership information is protected because the adversary cannot distinguish original tuples from fake tuples. Since the main focus of this paper is attribute disclosure, we do not intend to propose a comprehensive analysis for membership disclosure protection. We use the standard SVD-based prediction method.³ As in Netflix Prize, prediction accuracy is measured as the rooted-mean-square-error (RMSE). We compare slicing against the baseline method.

The baseline method will simply predict any user's rating on a movie as the average rating of that movie. Intuitively, the baseline method considers the following data publishing algorithm: the algorithm releases, for each movie, the average rating of that movie from all users. The baseline method only depends on the global statistics of the data set and does not assume any knowledge about any particular user.

An ongoing problem in Database As a Service (DAS) is how to increase the efficiency of retrieving encrypted data from remote untrusted servers without compromising security[2]. Bucketization is a privacy preserving technique for executing SQL queries over encrypted data on a DAS server. Bucketization partitions encrypted attributes into queryable tables (buckets), thereby disguising which records are requested. While a number of bucketization techniques are optimized for uniform query access, many Internet and private network access patterns reflect a non-uniform or Zipf-like trend. [3], [4], [5].

If query access is non-uniform, existing techniques may be subject to substantial performance degradation [6]. In order to evaluate that possibility, this thesis presents new bucketization technique, Query Sensitive Bucketization (QSB) that capitalizes on the probability distribution of non-uniform queries.

Two existing uniform bucketization techniques were implemented to (1) evaluate their performance when the distribution of queries is non-uniform, and (2) evaluate their performance relative to QSB. Among the measures used for performance analysis, a new security metric is presented, which quantifies the risk of an adversary estimating the true value distribution of an encrypted data store. Unlike existing security metrics, the new metric expresses information disclosed by the pattern of query access over an encrypted bucket set. Results showed that QSB improves query efficiency over uniform techniques, while maintaining a high level of data security.

QSB is not only an efficient example of query-based bucketization for DAS, but a conceptual model for future research, in which data are organized to accommodate a variety of query access patterns, thereby improving query efficiency and database security.

4. SCREENSHOTS

4.1 Admin Login Page

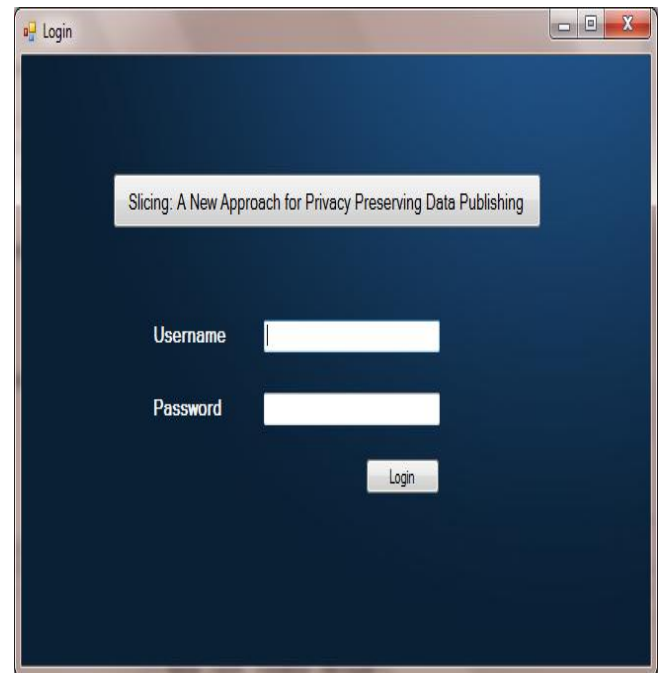


Fig-2: Admin Login Page

4.2 Creating the Dataset

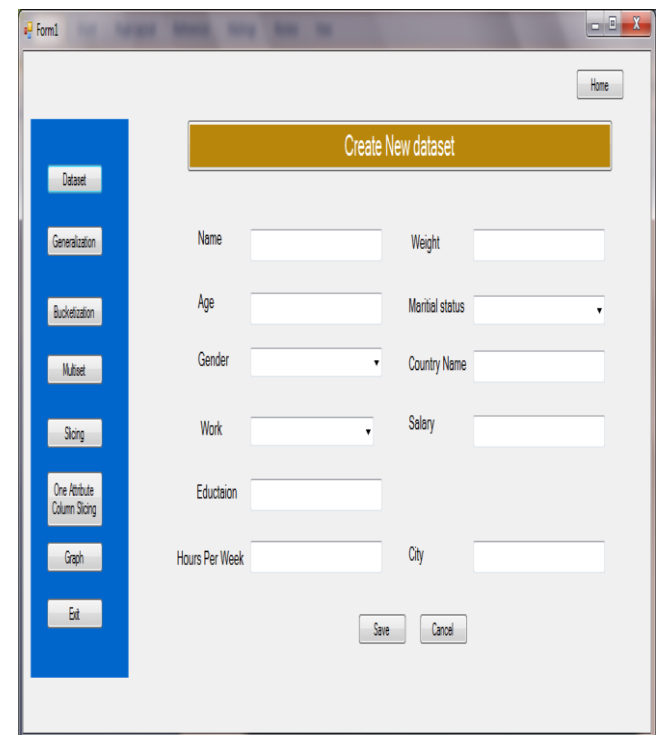


Fig-3: Creating the Dataset

4.3 Generalization Process

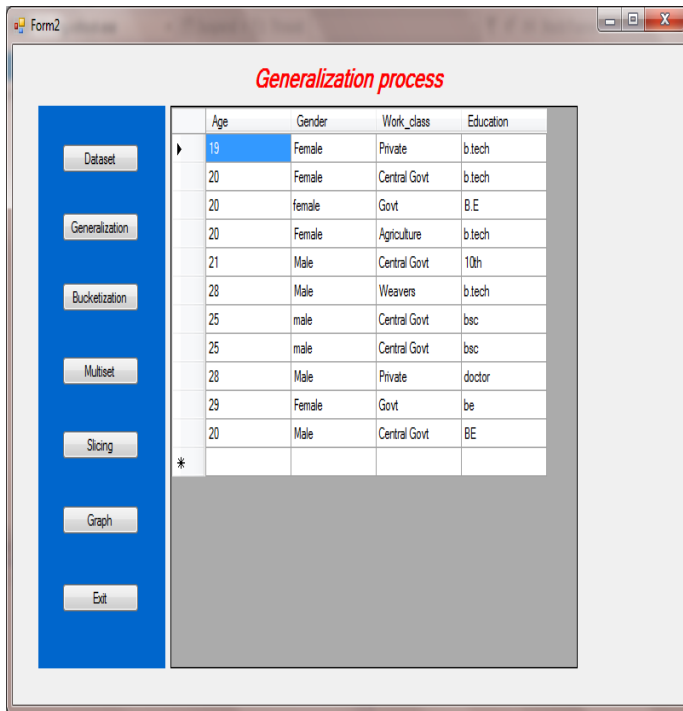


Fig-4: Generalization Process

4.5 Bucketization Process after Entering the Values

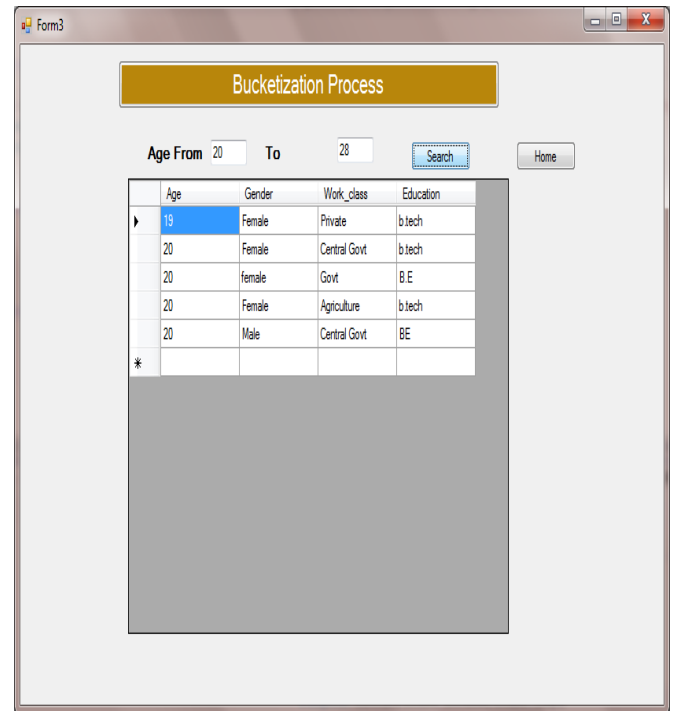


Fig-5: Bucketization process after entering the values

4.4 Bucketization Process before Entering the Values

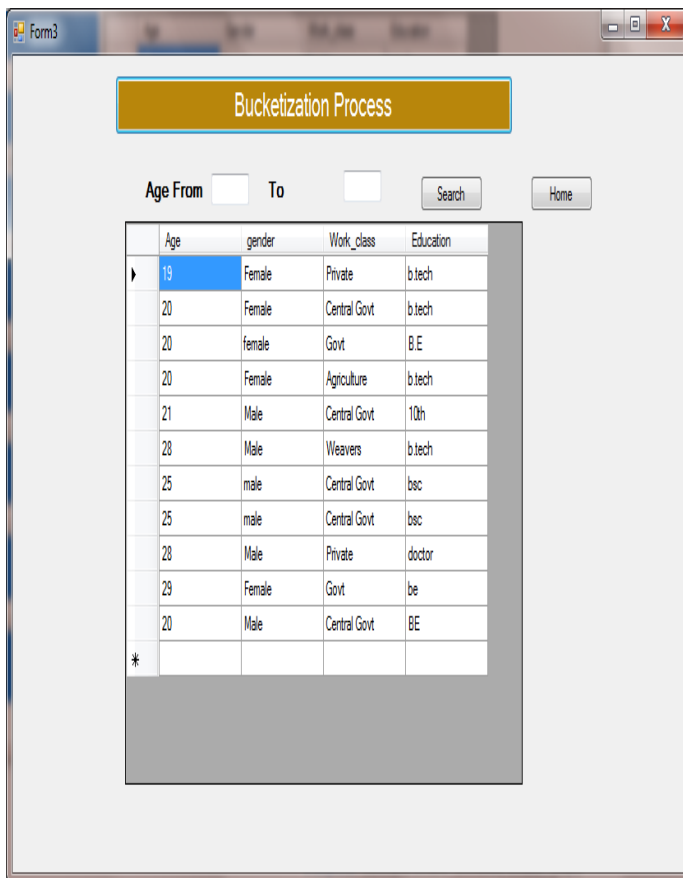


Fig-4: Bucketization process before entering the values.

4.6 Multiset Generalization

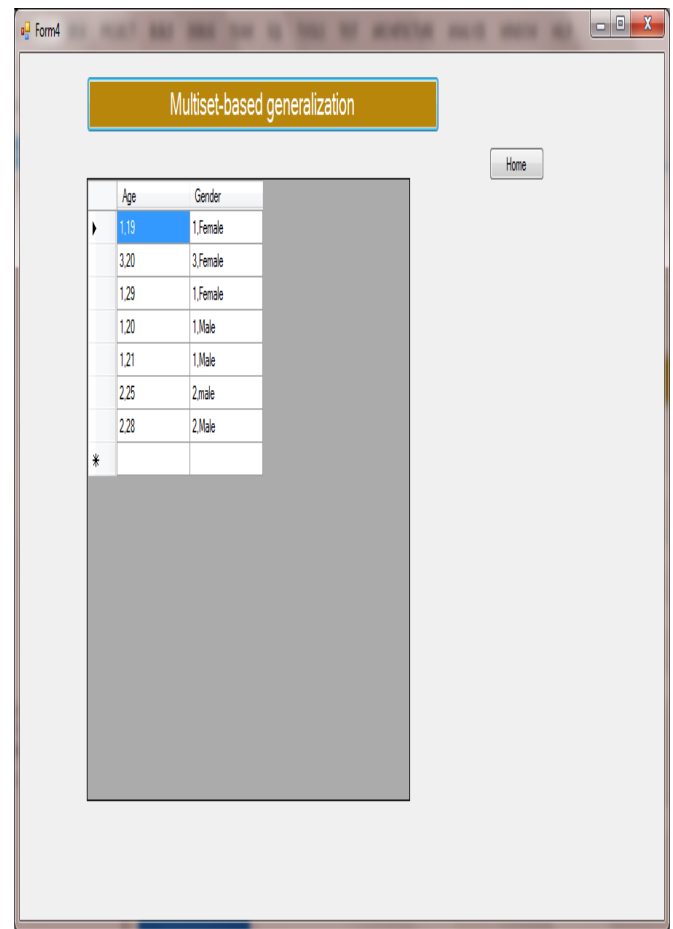
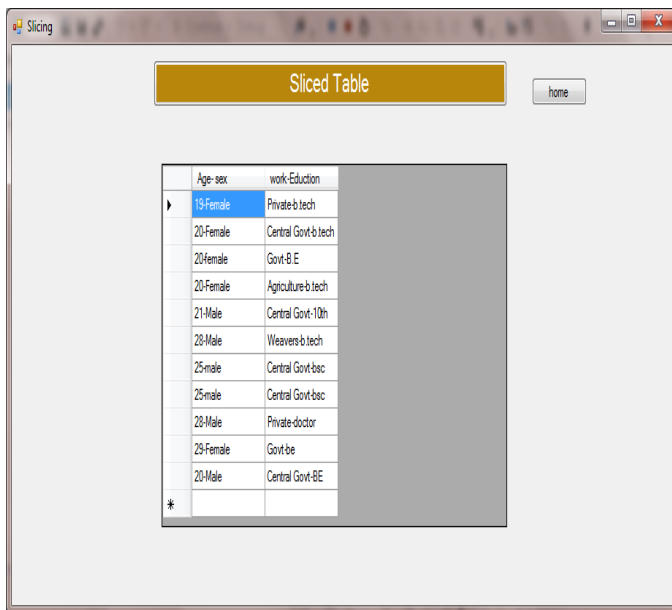


Fig-6: Multiset Generalization

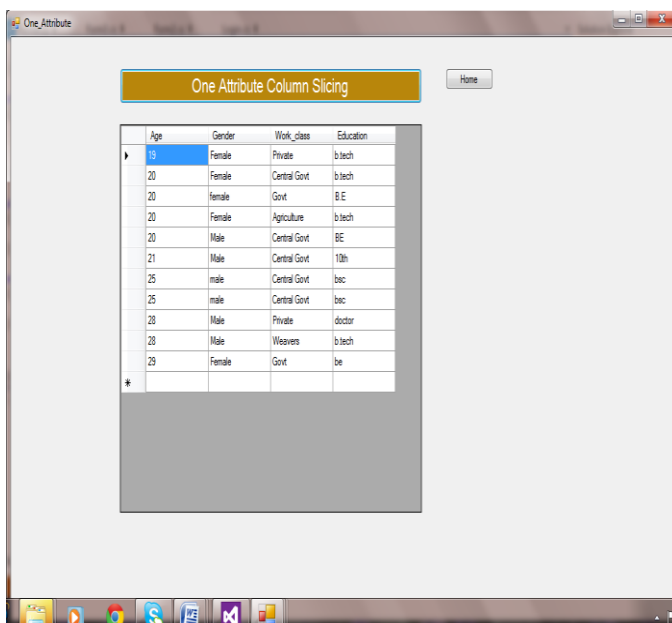
4.7 Sliced Table



Age	sex	work-Education
19	Female	Private-b.tech
20	Female	Central Govt-b.tech
20	Female	Govt-B.E
20	Female	Agriculture-b.tech
21	Male	Central Govt-10th
28	Male	Weavers-b.tech
25	male	Central Govt-bsc
25	male	Central Govt-bsc
28	Male	Private-doctor
29	Female	Govt-be
20	Male	Central Govt-BE
*		

Fig-7: Sliced Table

4.8 One Attribute Column Slicing



Age	Gender	Work_class	Education
19	Female	Private	b.tech
20	Female	Central Govt	b.tech
20	Female	Govt	B.E
20	Female	Agriculture	b.tech
20	Male	Central Govt	BE
21	Male	Central Govt	10th
25	male	Central Govt	bsc
25	male	Central Govt	bsc
28	Male	Private	doctor
28	Male	Weavers	b.tech
29	Female	Govt	be
*			

Fig-8: One Attribute Column Slicing

5. CONCLUSIONS

Thus this paper is used to transact the large data from the database in an easy and efficient way so that there is no data loss. And the large data can be combined together in order to reduce the space and transaction of the data are done in a faster way using the slicing. In this paper we mainly focus on the tuple partitioning that is only the particular rows can be grouped in order to combine the tables and generate the result. In future we can use the column generalization, in which particular column can be grouped accordingly. And also the column generalization may result in information loss.

REFERENCES

- [1]. R. Chan, Q. Yang, and Y. Shen, "Mining High Utility Itemsets," Proc. IEEE Third Int'l Conf. Data Mining, pp. 19-26, Nov. 2003.
- [2]. V.S. Tseng, C.-W. Wu, B.-E. Shie, and P.S. Yu, "UP-Growth: An Efficient Algorithm for High Utility Itemsets Mining," Proc. 16th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD'10), pp. 253-262, 2010.
- [3]. B.-E. Shie, H.-F. Hsiao, V., S. Tseng, and P.S. Yu, "Mining High Utility Mobile Sequential Patterns in Mobile Commerce Environments," Proc. 16th Int'l Conf. Database Systems for Advanced Applications (DASFAA '11), vol. 6587/2011, pp. 224-238, 2011
- [4]. M.Y. Eltabakh, M. Ouzzani, M.A. Khalil, W.G. Aref, and A.K. Elmagarmid, "Incremental Mining for Frequent Patterns in Evolving Time Series Databases," Technical Report CSD TR#08-02, Purdue Univ., 2008. 9
- [5]. J.H. Chang, "Mining Weighted Sequential Patterns in a Sequence Database with a Time-Interval Weight," Knowledge-Based Systems, vol. 24, no. 1, pp. 1-9, 2011.
- [6]. C.F. Ahmed, S.K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, "Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases," IEEE Trans. Knowledge and Data Eng., vol. 21, no. 12, pp. 1708-1721, Dec. 2009.

BIOGRAPHIES



Mrs.V.Suganthi received her B.E. in CSE from Jayaram college of Engineering and Technology, Bharathidasan university and her M.Tech in IT from Sathyabama University. She has got one year of Industry Experience. She is currently working as an Associate Professor in the department of Information Technology, IFET College of Engineering, Villupuram, India. She has published one International Journal. Her areas of interests includes Computer Networks, Programming Paradigms, Network Security.



Ms.J.Kalai vani received her B.E in CSE from VRS College of Engineering and Technology, Villupuram and M.Tech in CSE from Manonmaniam Sundaranar University. She is currently working as an Assistant Professor in the Department of Information Technology, IFET College of Engineering, Villupuram, India. She has published a book on Computer Graphics. She has published three papers in international journals. Her area of interests includes Computer Networks, Cryptography and Network Security, Computer Graphics.