ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 10, October 2013

# Fundamental methods to evaluate horizontal aggregation in SQL

**Krupali R. Dhawale 1, Vani A. Hiremani 2**

Abstract— **In data mining, we are extracting data from historical knowledge and create data sets. Many hyper graph concepts and data mining algorithms are used to create datasets in tabular format which consist complex queries, joining tables and aggregated functions like min, sum, count, max etc . This task is very hard and time taking. Tabular format takes suitable input to prepare data sets. But, Existing SQL aggregations having limited capacity to prepare data sets. They go back one column per aggregated group.**

**In this paper, we propose fundamental methods to evaluate horizontal aggregations to represent an outline generate SQL code to return in a horizontal tabular format by using SPJ, CASE and PIVOT methods. This class of new function is called as horizontal aggregation.**

**This paper consist three methods those are SPJ, CASE and PIVOT method to evaluate horizontal aggregations. PIVOT and UNPIVOT, these are two operators are used to exchange rows and columns that appears the data transformations helpful to create data analysis, data modeling and data visualizations of user records. Classification is one of the most significant tasks performed in Data Mining applications. This paper presents an efficient SQL implementation of the C4.5 algorithm to perform classification in very large prepared databases. Our version can effectively handle high dimensional data records by using C4.5 algorithm and make decisions. We expect this work to be useful for data mining programmers and users who want to classify and group the large data sets inside a relational DBMS. However, if the tree obtained is very large (a lot of nodes and leaves) then they are less comprehensible. A decision tree can be directly altered into a set of IF-THEN rules that are one of the most well-liked forms of knowledge illustration, due to their simplicity and comprehensibility.**

***Index Terms*- aggregations, SQL, pivoting, data preparations.**

## I. INTRODUCTION

Data mining sometimes called data or knowledge discovery is the process of extracting data from different perspectives and aggregate data is useful for future scope.

***Krupali R. Dhawale, ME student,*** *Alard College of engg. And management, Marunje, Pune, pune University . Pune, India*

***Vani A. Hiremani, Assistant Professor of*** *computer and science Department of Alard College of engg. And management, Pune, India*

Data mining essential is one of a number of analytical tools for applied to data. It allows users to analyze data from many different databases and categorize it, and summarize the relationships identified.

Generally, data mining is the process of searching patterns or correlations among relational databases repository of fields in large databases where data sources can include database, data warehouse, web, information repositories etc. Data mining turns a huge collection of data into knowledge for global challenges. It is nothing but 'knowledge mining data" which is somewhat long and similar to knowledge extraction, data/pattern analysis, data dreading.

Every study area uses unusual terms to illustrate a data set. In data mining the regular terms are point-dimension. Statistics literature use observation variable. Machine learning research uses instance-feature. This paper consist three methods those are SPJ, CASE and PIVOT method to evaluate horizontal aggregations. PIVOT and UNPIVOT, these are two operators are used to exchange rows and columns that appears the data transformations helpful to create data analysis, data modeling and data visualizations of user records. Classification is one of the most significant tasks performed in Data Mining applications. This paper presents an efficient SQL implementation of the c4.5 algorithm to perform classification in very large databases. Our version can effectively handle high dimensional data records by using C4.5 algorithm.

To complete the study of mined databases on spreadsheets, it may be more flexible to having the aggregated functions on single group on single row. For example make student id information, new students admission record, test marks, result, chart or contrast datasets with recurring information. OLAP tools make SQL codes which modify results more efficiently on aggregation and transposition mechanisms. These horizontal aggregations construct extra features of standard SQL aggregations, which return a set of values in a horizontal layout, in its place of scalar standards.

There are some advantages for horizontal aggregation. First one is horizontal aggregation correspond to a pattern to make SQL code from a data mining tool. This SQL code reduces physical work in the data preparation part in data mining related project. Second is automatically generated code, which is well-organized than end user written SQL code. Thus datasets for the data mining projects can be

produced in less time. Third advantage is the data sets can be produced completely within the DBMS. C4.5 algorithms are used to generate the decision tree of the given dataset, that dataset is the result of horizontal aggregation. Next part presents the decision tree of the given aggregated dataset and different methods existing for aggregation and Conclusion

## II. RELATED WORK

M. Madhavi and S. Kavitha, [1] Experiments with large tables compare the proposed query evaluation methods. CASE method has similar speed to the PIVOT operator and it is much faster than the SPJ method. The CASE and PIVOT methods exhibit linear scalability, whereas the SPJ method does not. Rajesh Reddy Muley, Sravani Achanta and Prof.S.V.Achutha Rao, in [2] explains us the way to use the data mining methods to show the datasets by mining the data from different tables at the same time. The methods which are suitable for data mining analysis are CASE, SPJ and PIVOT. Coming with CASE they show two possibilities i.e. Vertical view and also the Horizontal View. This paper thus satisfies the main concern i.e. reducing the overload on the databases for retrieval of data. Durka.C and Kerana Hanirex.D in [3] achieves horizontal aggregations through some constructs built that includes SQL queries as well. In the proposed system, a new standard of pivoting option is incorporated using Data mining. This can be achieved with the tool SAAS (SQL Server Analysis Services). Data will be taken and transformed into knowledge cubes. This can be achieved with MDX (Multi-Dimensional expression) queries. On top of that, the knowledge data will be customized based on "Generalized and Suppression Algorithm" .Its provide privacy for dataset. The data will be taken and it will be transformed into knowledge cubes. This can be achieved with MDX (Multi Dimensional eXpression) queries. On top of that, the knowledge data will be customized based on "Generalized and Suppression Algorithm". In addition to this, the performance efficiency among case, SPJ and pivot methods will be analyzed. Conor Cunningalam [4] proposed an optimization and implementation strategies in an RDBMS which uses two operators i.e., PIVOT and UNPIVOT operators on tabular data that return rows and columns, enable data transformation useful in data modeling, data psychoanalysis, and data appearance . Pivot is an accumulation of Group By with unique limits and optimization opportunities, and this makes it very easy to begin incrementally on peak of existing situation implementations. K. Anusha, P. Radhakrishna and P. Sirisha with large tables show our proposed horizontal aggregations evaluated with the CASE method have similar performance to the built-in PIVOT operator. This is remarkable since their proposal is based on generating SQL code and not on internally modifying the query optimizer. Both CASE and PIVOT evaluation methods are significantly faster than the SPJ method. Precomputing a cube on selected dimensions produced acceleration on all methods [6]. C. Ordonez [7] introduces vertical and horizontal aggregations functions. Vertical aggregations revisit one row for each percentage in vertical form like normal SQL aggregations. Horizontal aggregations returns each set of percentages adding 100% on the same row in horizontal layout. This paper presents three horizontal aggregations methods CASE, PIVOT and SPJ. CASE is based on the SQL CASE construct, PIVOT makes use of built in pivoting facility in SQL while SPJ uses standard SQL aggregations. Data mining is motivated by the decision support problem faced by most large retail organizations. A record in organizations login detail application for the user, Transaction date and the payment dates in the application. Horizontal aggregations consequences in large volumes of data sets which are then partition into fixed clusters are very important in the system. This can be performed by k- means clustering algorithm [8].

This paper [9] horizontal aggregations can be used as a database method to automatically generate efficient SQL queries with three sets of parameters: grouping columns, sub-grouping columns and aggregated column. The fact that the output horizontal columns are not available when the query is parsed (when the query plan is explored and chosen) makes its evaluation through standard SQL mechanisms infeasible. Our experiments with large tables show our proposed horizontal aggregations evaluated with the CASE method have similar performance to the built-in PIVOT operator. In a clustering algorithm is explored which makes use of SQL queries internally. It is capable of viewing horizontal layout for further mining operations. SQL extensions to define aggregate functions for association rule mining. Their optimizations have the point of avoiding joins to correspond cell formulas, but are not optimized to achieve limited transposition for each group of effect rows. H. Wang. C. Zaniolo [10] proposed a small but Complete SQL addition for data Mining and Data Streams. This technique is a great database language and system that enables users to increase absolute data-intensive applications in SQL by script new aggregates and table functions in SQL, rather than in practical languages as in present Object-Relational systems. Since usual query graph models are too little for modeling outer join queries with difficult predicates, they in progress the desired hyper graph abstraction and algorithms for reordering such queries with joins and outer joins. As a consequence, the query optimizer can discover a considerably larger space of finishing plans, and choose one with a low cost. Further, these algorithms are easily incorporated into well known and widely used enumeration methods such as dynamic programming [11].

## III. AGGREGRATION

Horizontal aggregations propose a new class of functions that aggregate numeric expressions and the result are transposed to produce data sets with a horizontal layout. The operation is needed in a number of data mining tasks, such as unsupervised classification and data summation, as

well as segmentation of large mixed data sets into smaller uniform subsets that can be easily manage, separately model and analyzed. To create datasets for data mining related works, efficient and summary of data are needed. Database as their nature contains large amount of data. To extract information from the database Structured Query Languages are used. SQL commonly used for the aggregation of large volumes of data. With the help of aggregation details in one table can be aggregated with details in another table. Aggregation functions play a major in the summarization of tables. Normal SQL aggregation functions are sum ( ), avg ( ), min ( ), max ( ) and count ( ).

➢ *Vertical aggregations*:

Vertical aggregation is similar to standard SQL aggregations. This produces results in a vertical format and contains more rows. There are some approaches which produce results in vertically aggregated form.

➢ *Horizontal aggregations:*

Horizontal aggregations are also similar to standard SQL aggregations but this can produce results in horizontal tabular format. Here data sets for all the operations are produced from some data mining tool and apply the aggregation operations on that dataset[1]. To produce results in horizontal layout small syntax extensions to normal SQL syntax is needed.

The syntax for horizontal aggregation is given below.
*SELECT columns, Aggregation Function*
*(Measure column BY Aggregating Parameters)*
*FROM GROUPING columns*

## IV. PROPOSED METHODOLOGY

### I. *SQL code generation*

For that this proposed system collect particular needed attributes from the different fact tables and displayed columns in order to create data in horizontal layout. Main aim is to classify a pattern to generate SQL code combine aggregation and transposition (pivoting)[1].

A second aim is to expand the SELECT statement with a clause that combines transposition with aggregation. Consider the following GROUP BY query in regular SQL +that takes a subset L1. . . Lm from D1. . . D p: SELECT L1... Lm, sum (A) FROM F1, F2 GROUP BY L1, Lm;

In a horizontal aggregation there are four input parameters to generate SQL code:

1) The input table F1, F2……, Fn
2) The list of GROUP BY columns L1. . . Lj,
3) The column to aggregate (A),
4) The list of transposing columns R1. . . Rk. This aggregation query will produce a wide table with m+1 column (automatically determined), with one group for each unique combination of values L1. Lm and one aggregated value per group (i.e., sum (A)). This query estimate the query optimizer takes three input parameters. First parameter is the input table F. Second parameter is the list of grouping columns L1. Lm. And the final parameter is the column to aggregate (A) .

### II. *Query evolution method*

There are three query evolution methods.

- ❖ SPJ method
- ❖ PIVOT method
- ❖ Case method
- ❖ *SPJ method*

SPJ method based on the relational operators and it is very important to theoretical point of view. Then all such tables are joined in order to generate a table containing horizontal aggregations. The actual implementation is based on the details given in data sets.

Proposed syntax is as follows.

SELECT ($L_1 \ldots L_j$), H (A BY $R_1, \ldots, R_k$)
FROM F
GROUP BY ($L_1 \ldots L_j$);

❖ *Pivot method*

PIVOT transforms a series of rows into a series of fewer numbers of rows with additional columns. Data in one source column is used to determine the new column for a row, and another source column is used as the data for that new column.

❖ *Case method*

CASE method can be performed by combining GROUP-BY and CASE statements [9]. It is more efficient and has wide applicability. CASE statement evaluates the Boolean expression and return value from the selected set of values. CASE statement put the result to NULL when there is no matching row is found. This also produce resultant table in a horizontal layout.

CASE programming construct available in SQL. It is most evaluation method and it can be evaluated by GROUP-BY and CASE statement. This will help us to minimize a lot of space used by user details. CASE method can be seen in the horizontal aggregation.

SELECT columns, Aggregate Function
(CASE WHEN Boolean expression THEN result ELSE result expression END)
FROM table GROUP BY column.

## V. DATASET PREPRATION

Dataset preparation is very important for any of the operations in the data mining analysis. Preparation of dataset addresses many issues and there are solutions for overcoming this problem. For performing operations on data stored inside the database system, users normally use SQL queries to retrieve those data. After retrieving perform various extractions and transformations on the dataset to make them suitable for application. Some approaches require demoralized table than normalized table. Because that contain more details than normalized tables and many

analysis require analysis on large amount of data. There are four steps for the dataset preparation. Dataset preparation starts with data selection. In data selection, the analyst wants to perform analysis on the available data and select appropriate data for analysis. Second step is data integration. In data integration, data collected from different source are combined and stored inside a table. Third one is the data transformation. In data transformation the analyst wants to transform data into the format required for each operation. The last step is the data reduction. Here the data is compressed for the easiness of the analysis. Creating selected the dataset which is applicable for C4.5 algorithm and implements the C4.5 algorithm by using Weka tool.

These extensions represent a synthesis of many ideas from the published work in semantic modeling plus the introduction of new rules for insertion, update, and deletion, as well as new algebraic operators.

The major problems in the creation and transformation of variables for analysis are selection of appropriate record from the available large dataset and preparation of efficient SQL queries to optimize them. Most the issues in the creation and transformation of dataset are related to summarization, aggregation, demoralisation and cross-tabulation. Sometimes analysis need summarized details. So there is a need for summarized data. Cross-tabulation is also an important concept because it gives detailed information after analysis in a horizontal tabular format. Those are easy to understand than vertical format. Horizontal aggregation almost similar to normal aggregation but it uses some syntax extensions. Selecting SQL queries face some difficulties when using left outer join for effectiveness and use of appropriate SQL queries for each operation and handling of multiple primary keys. By using three fundamental methods to evaluate horizontal aggregation in SQL.

There is a data preparation framework for efficiently preparing dataset for analysis. Create the dataset by update, delete, insert, and modify SQL queries. Firstly create data sets by using SPJ method then create logging application user detail form in which insert record, delete record, add new entry, user information, maintain a data records rate an admin module, login module. C4.5 algorithm builds decision trees and classify given data sets of training data. Aggregate the data summarize the data decision trees are considered easily understood models because a reasoning process can be given for each conclusion. However, if the tree obtained is very large (a lot of nodes and leaves) then they are less comprehensible. A decision tree can be directly transformed into a set of IF-THEN rules that are one of the most popular forms of knowledge representation, due to their simplicity and comprehensibility. So, C4.5 algorithm is simple for instructors to understand and interpret. Quality of any analysis depends on the quality of data being processed. Since dataset preparation is most expensive and time consuming task, this dataset preparation is very important. The data is stored inside the database system can get the benefit of database management system..

These extensions represent a synthesis of many ideas from the published work in semantic modeling plus the introduction of new rules for insertion, update, and deletion, as well as new algebraic operators.

## VI. Structured Query Language

Creation and transformation of dataset is related to summarization, aggregation, demoralization, and cross-tabulation. Sometimes analysis need summarized details then detailed. So there is a need for summarized data. Cross-tabulation is also an important concept because it gives detailed information after analysis in a horizontal tabular format. Those are easy to understand than vertical format. Horizontal aggregation almost similar to normal aggregation but it uses some syntax extensions. Selecting SQL queries face some difficulties when using left outer join for effectiveness and use of appropriate SQL queries for each operation and handling of multiple primary keys.

The major problems in the creation and transformation of variables for analysis are selection of appropriate record from the available large dataset and preparation of efficient SQL queries to optimize them. Most the issues in the creation and transformation of dataset is related to summarization,aggregation,demoralization,
ross-tabulation. Sometimes analysis need summarized details then detailed. So there is a need for summarized data. Cross-tabulation is also an important concept because it gives detailed information after analysis in a horizontal tabular format. Those are easy to understand than vertical format. Horizontal aggregation almost similar to normal aggregation but it uses some syntax extensions. Selecting SQL queries face some difficulties when using left outer join for effectiveness and use of appropriate SQL queries for each operation and handling of multiple primary keys.

## VII. Experimental Result and Analysis

In this paper we create login application for the user. Firstly, we create admin login form, search new connection form, user login form. Admin login form consist user bill details. User bill details can be modified and aggregated by SPJ method, CASE method and PIVOT method. PIVOT method can be transposing the rows into aggregated columns. In this paper, upload the new connection form and extract the database manage the user details files. This paper, helps for login the user account ,search connection form ,download the new connection form, change the password , see the bill details. All this information aggregated in horizontal tabular format by SPJ, CASE and PIVOT wise method. We believe this is remarkable paper, since our proposal is based on generating SQL code and not on internally modifying the query optimizer. Both CASE and PIVOT evaluation methods are significantly faster than the SPJ method. C4.5 algorithm is used to classify the given prepared relational database. C4.5 is a well known

algorithm used for classifying datasets. It includes decision trees and rules from datasets, which could contain definite and numerical attributes. The rules could be used to predict definite values of attributes from new records. This paper discusses a summary of data classification and its techniques, the essential methods of C4.5 algorithm, the methods and study of the results of the research, which utilizes C4.5 for Classifying user details. C4.5 performs well in classifying the data set, but more data needs to be composed in order to enhance supportive rules.

## VIII. COMPARATIVE STUDY

Various approaches for performing horizontal aggregation are taken for comparison is as follows.

❖ *Grouping combinations*:

This operator is developed to handle the aggregation and grouping of high dimensional data. This operator can solve limitations of normal GROUPING operator. The operators like GROUPING SET, ROLLUP, and CUBE can also perform aggregation and can produce tabular results. But these are difficult to use when the available input dataset is very large. When the available input dataset is large GROUPING SET operator require long complex SQL queries. The ROLL UP operator can perform aggregation on smaller datasets and produce tabular results vertical format. But the vertical format is not efficient for many data mining approaches. The CUBE operator can perform aggregations on large datasets. But the CUBE operator eliminates some of the details when aggregation is performed. Because of these limitations GROUPING COMBINATION operator is developed. But the GROUPING COMBINATION operator can implemented only with the help of complex algorithms. So its performance is low in the case of execution.

❖ *Atlas:*

It is a database language developed to solve the limitations of SQL operator. ATLaS [6] can perform aggregations that are not possible with standard SQL. Standard SQL can support only basic aggregation operations. This language use aggregations and table functions in SQL. To perform operations in ATLAS entire SQL statement is divided into three functions INTIALIZE, ITERATE, TERMINATE. Declarations are given in the INITIALIZE section. The major operation is specified in the ITERATIVE section. The final statement to execute is specified in the TERMINATE sections. The major advantage of ATLaS is that it can support online aggregations. In online aggregation user evaluate aggregation query in an online fashion execution. But the execution of ATLaS operator consumes more space than executing with normal SQL [6]. Also it cannot results in horizontal tabular format.

❖ *Horizontal and vertical percentage:*

This aggregations help to calculate percentages for operations using vertical and horizontal aggregations. Vertical percentage aggregation returns one row for percentage in vertical format. Horizontal percentage aggregation returns entire 100% of results on the same row. This percentage aggregation used only for computing percentages in vertical or horizontal format. These aggregations are similar to normal vertical and horizontal aggregation except that it can compute results only in percentage format. So there may be extra work in the percentage conversion when other computations are required on the dataset.

TABLE I
Comparison of aggregation approaches

| Method discussed | Types of aggregation | Feature |
|---|---|---|
| Grouping combination operator | Horizontal and Vertical | Implemented with complex algorithms |
| Atlas | Vertical Aggregation | Solve limitation of normal SQL |
| Vertical and horizontal percentage aggregation | Vertical and horizontal aggregation | Can only operate on percentages |
| Interpreted storage format | Vertical and horizontal aggregation | Data retrieval is difficult |
| UNPIVOT operator | horizontal aggregation | Use small syntax extension in select statement |

❖ *Interpreted Storage Format :*

This is developed to handle null values in horizontal and vertical layouts. Interpreted format can handle all the sparse data management complexities. Horizontal aggregation requires more space due to large number of null values. Vertical aggregations have small number of null values. Interpreted storage format store nothing for null attributes. When the tipple has value for an attribute in the table, attribute identifier (attribute_id), a length field, value appears in the tuple . This stored along with particular head. The major problem here is that the value stored in this format is not easily accessible for operations.

❖ *UNPIVOT Operator:*

UNPIVOT operator is opposite of PIVOT operator that is they transform columns into rows. This creates additional rows from columns to produce a big table. Because of these

vertical layout it cannot use for most of the mining algorithms which require horizontal table as input. UNPIVOT operator [9] is commonly used for the statistical computation of some data mining approaches. The normal syntax is given below.

SELECT columns FROM table UNPIVOT

(Measure Column FOR Pivot Column IN (Pivot Column Values))

## IX. Conclusion

In future experiments, we want to determine the compressibility of each classification model and use prepared data with more information about the login user (i.e. account , new connection form , upload the form and bill details of users ) and of higher quality (complete data about user bill account that have done all the activities). In this way we could measure how the quantity, attributes and quality of the data can affect the performance of the use of prepared data and make a decision tree among those entire prepared database using C4.5 algorithms.

C4.5 algorithm builds decision trees and classify given data sets of training data. Aggregated data summarize the data and decision trees are considered easily understood models because a reasoning process can be given for each conclusion. However, if the tree obtained is very large (a lot of nodes and leaves) then they are less comprehensible. A decision tree can be directly altered into a set of IF-THEN rules that are one of the most well-liked forms of knowledge illustration, due to their simplicity and comprehensibility. So, C4.5 algorithm is simple for instructors to understand and interpret and implement that by Weka tool.

## Acknowledgment

I would like to express my sincere thanks to my Guide **Vani A. Hiremani, Assistant professor Of Alard college of engg. And management , Pune** for their consistence support and valuable suggestions.

## References

[1] M. Madhavi and S. Kavitha," Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis", *Madhavi et al. / IJEA, Vol. 1 Issue 6 ISSN: 2320-0804,* PP 1-7, 2013.

[2] Rajesh Reddy Muley, Sravani Achanta and Prof.S.V.Achutha Rao, "Query Optimization Approach in SQL to prepare Data Sets for Data Mining Analysis", *International Journal of Computer Trends and Technology (IJCTT) – volume4Issue8*,pp 1-5,August 2013.

[3] Durka.C and Kerana Hanirex.D, '' An Efficient Approach for Building Dataset in Data Mining", *International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3*,pp 1-5, March 2013.

[4] C. Ordonez, "Horizontal Aggregations for Building Tabular Data Sets," *IEEE Trans. Knowledge and Data Eng*," VOL. 24, NO. 4, pp. April 2012.

[5] Pradeep Kumar and Dr. R. V. Krishna, "Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis," *IOSR Journal of Computer Engineering (IOSRJCE)* ISSN: 2278-0661, ISBN: 2278-8727 Volume 6, Issue 5, PP. 36-41, (Nov. - Dec. 2012).

[6] K. Anusha, P. Radhakrishna and P. Sirisha," Horizontal Aggregation using SPJ Method and Equivalence of Methods", *IJCST, Vol. 3, Issue 1, Spl. 5*, pp 1-4, Jan. - March 2012 .

[7] Carlos Ordonez and Zhibo Chen," Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis", *IEEE transactions on knowledge and data engineering, vol. 24*, NO. 4, pp 1-14, APRIL 2012.

[8] Nisha.S and B.Lakshmipathi, "Optimization of Horizontal Aggregation in SQL by Using K-Means Clustering", *International Journal of Advanced Research in Computer Science and Software Engineering* , Volume 2, Issue 5, ISSN: 2277 128X, PP. 1-6, May 2012.

[9] Mr. Ranjith Kumar K and Mrs. Krishna Veni," Prepare datasets for data mining analysis by using horizontal aggregation in SQL", *Ranjith Kumar K et al, Int.J. Computer Technology & Applications,* Vol 3(6), 1945-1949 IJCTA, pp. 1-5, Nov-Dec 2012.

[10] Haixun Wang, Carlo Zaniolo and Chang Richard Luo," ATLaS: a Small but Complete SQL Extension for Data Mining and Data Streams"**,** *Proceedings of the 29th VLDB Conference***,** Berlin, Germany, pp. 1-4, 2003.

[11] P. Goel, and B.R. Iyer, "Hyper graph Based Reordering of Outer Join Queries with Complex Predicates," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '95) , pp. 304-315, 1995.

**First Author**

**Krupali Rupesh Dhawale** received her B.E. degree from M.I.E.T. Gondia, Nagpur University, and Maharashtra state, India in the year 2008. Currently, Pursuing M.E. in Computer Engineering from Alard College of engineering and management, Marunje, Pune, Maharashtra state, India. Her research interests include in Data mining, Mobile computing.

**Second Author**

**Vani A. Hiremani, Assistant professor,** received the bachelor's degree in Computer Science & engg. From Hirasagar Institute of tech., Nidasoshi, Visvesvaraya technological university, Belgaum, Karnataka state, India Master degree in computer science and engg. From Basaveshwar engg. College Bagalkot, Karnataka state, India, Her research interests include in biometrics, image processing, pattern regonation computer security and networking.