

Cloud Framework for Association Rule Hiding

*Thesis submitted in partial fulfilment of the requirements for the award
of degree of*

**Master of Software Technology
in
Computer Science and Engineering**

Submitted By
Peeyush Varshney
Roll No.
2K14/ SWT/509

Under the supervision of
Manoj Kumar
Associate Professor



Dec 2017

**Department of Computer Engineering
Delhi Technological University
Delhi – 110042**

CERTIFICATE



Delhi Technological University
(Government of Delhi NCR)
Bawana Road, New Delhi-42

This is to certify that the thesis entitled “**Cloud Framework for Association Rule Hiding**” done by me for the Major project for the award of degree of **Master of Technology** Degree in **Software Technology** in the **Department of Computer Engineering**, Delhi Technological University, New Delhi is an authentic work carried out by me under the guidance of **Prof. Manoj Kumar**.

Signature:

Student Name
Peeyush Varshney
2K14/SWT/509

Above Statement given by Student is Correct.

Project Guide:

Manoj Kumar
Associate Professor
Department of Computer Engineering
Delhi Technological University, Delhi

Acknowledgement

No volume of words is enough to express my gratitude towards my guide **Manoj Kumar**, Department of Computer Science & Engineering, Delhi Technological University, Delhi, who has been very concerned and has aided for all the materials essentials for the preparation of this thesis report. He has helped me to explore this vast topic in an organized manner and provided me all the ideas on how to work towards a research-oriented venture.

I am also thankful to **Dr. Rajni Jindal**, Head of Computer Science & Engineering Department and **Dr. Ruchika Malhotra**, Coordinator, for the motivation and inspiration that triggered me for the thesis work.

I would also like to thank the staff members and my colleagues who were always there at the need of hour and provided with all the help and facilities, which I required, for the completion of my thesis work.

Most importantly, I would like to thank my parents and the almighty for showing me the right direction unexpected, to help me stay calm in the oddest of the times and keep moving even at times when there was no hope.

Peeyush Varshney
(2K14/SWT/509)

ABSTRACT

Data mining techniques are the result of a long process of research and product development. This evolution began when business data was first stored on computers, continued with improvements in data access, and more recently, generated technologies that allow users to navigate through their data in real time. APRIORI algorithm, a popular data mining technique and compared the performances of a linked list based implementation as a basis and a tries-based implementation on it for mining frequent item sequences in a transactional database. In this report, I examine the data structure, implementation and algorithmic features mainly focusing on those that also arise in frequent item set mining. This algorithm has given us new capabilities to identify associations in large data sets. However, a key problem, and still not sufficiently investigated, is the need to balance the confidentiality of the disclosed data with the legitimate needs of the data users. One rule is characterized as sensitive if its disclosure risk is above a certain privacy threshold. Sometimes, sensitive rules should not be disclosed to the public, since among other things, they may be used for inferring sensitive data, or they may provide business competitors with an advantage. Therefore, next I worked with some association rule hiding algorithms and examined their performances in order to analyse their time complexity and the impact that they have in the original database.

Table of Contents

List of Figures	vii
List of Acronyms	viii
List of Tables	ix
CHAPTER 1. INTRODUCTION	1
1.2 MOTIVATION	2
1.3 OBJECTIVE	2
1.4 CONTRIBUTION	3
1.5 ORGANISATION OF PROJECT REPORT	3
CHAPTER 2. THEORETICAL BACKGROUND AND LITERATURE SURVEY	4
2.1 ASSOCIATION RULE MINING	4
2.1.1 NEED FOR ASSOCIATION RULE MINING [2]	4
2.1.2 EXPLANATION WITH EXAMPLE	6
2.2 SECURE MULTIPARTY COMPUTATION [1]	7
2.2.1 FEASIBILITY OF SMC	8
2.3 DATA HIDING	9
2.4 HIDING TECHNIQUES	9
2.4.1 ACCURACY OF HIDING ALGORITHMS	10
2.4.2 THEORETICAL COMPARISON OF HIDING TECHNIQUES	10
CHAPTER 3. PROPOSED MODEL AND WORK	12
3.1 APPROACH TO ARM	12
3.1.1 FREQUENT ITEMSET GENERATION	12
3.1.2 RULE GENERATION	14
3.2 DATA HIDING TECHNIQUES	16
3.2.1 PROBLEM FORMULATION	16
3.2.2 RULE HIDING ALGORITHMS	17
3.3 PROPOSED CLOUD MODEL	22
3.3.1 STRENGTHS OF MODEL	23
3.3.2 USER INTERFACE	23
3.3.3 DATA INTEGRITY	26
3.4 SUMMARY	26
CHAPTER 4. SIMULATION AND RESULTS	28
4.1 HIDING ALGORITHMS ALONG WITH APRIORI	28
4.2 SIMULATION RESULT	29

4.2.1 RESULTS OF APRIORI ALGORITHM.....	29
4.2.2 RESULTS OF MDSRRC HIDING ALGORITHM.....	32
4.2.3 RESULTS OF ISLF HIDING ALGORITHM	34
4.3 PERFORMANCE EVALUATION OF MDSRRC AND ISLF.....	35
4.3.1 PERFORMANCE EVALUATION PARAMETERS	35
4.3.2 RESULTS OF PERFORMANCE EVALUATION	36
CHAPTER 5. CONCLUSION.....	40
REFERENCES	41

List of Figures

Figure 1: Structure of SMC.....	6
Figure 2: Frequent Itemset Generation in ARM	10
Figure 3: Pruned Structure of ARM.....	12
Figure 4: Pruned Structure of lattice of Rules in ARM	13
Figure 5: Proposed cloud model	22
Figure 6: Screenshots of the UI	23
Figure 6.a: Screenshot 1: Registration Page.....	23
Figure 6.b: Screenshot 2: Login Page.....	24
Figure 6.c: Screenshot 3: User Page.....	24
Figure 6.d: Screenshot 4: File Upload window	25
Figure 6.e: Screenshot 2: Algorithm Evaluation window	25
Figure 6.f: Screenshot 2: Integrity Check window.....	26
Figure 7: Screenshot 2: Dataset	28
Figure 8: Screenshot of Simulation Result	29
Figure 8.a: Screenshot 1: Pruned tree at Level 1	29
Figure 8.b: Screenshot 2: Pruned tree at Level 2.....	30
Figure 8.c: Screenshot 3: Pruned tree at Level 3.....	30
Figure 8.d: Screenshot 4: Pruned tree at Level 4.....	31
Figure 9: Results of Apriori	31
Figure 10: Mined Rules used in MDSRRC	32
Figure 11: Sensitive Rules defined by user.....	33
Figure 12: Sanitized Database after applying MDSRRC.....	33
Figure 13: New Mined Rules (MDSRRC).....	33
Figure 14: Mined Rules used for ISLF	34
Figure 15: Sanitized Database after applying ISLF	34
Figure 16: New Mine Rules (ISLF)	34
Figure 17: Graph of Dissimilarity vs Sensitive Rule	37
Figure 18: Graph of Artificial Pattern vs Sensitive Rule	37
Figure 19: Graph of Misses cost vs Sensitive Rule	38
Figure 20: Graph of Dissimilarity vs Support.....	38
Figure 21: Graph of Artificial Pattern vs Support.....	39
Figure 22: Graph of Misses cost vs Support.....	39

List of Acronyms

ARM: Association Rule Mining

SMC: Secure Multiparty Computation

ISLF: Increase Support of Left Hand Side

DSRF: Decrease Support of Right Hand Side

PPDM: Privacy Preservation in Data Mining

MDSRRC: Modified Decrease Support of Right hand side in Rule Cluster

List of Tables

Table 1: ARM Example	5
Table 2: Sample Database for Data Hiding Techniques	14
Table 3: Large Item set obtained from sample database.....	14
Table 4: Rules Derived from Table 3.....	14
Table 5: Sample Database that uses the proposed notation	14
Table 6: Performance Evaluation by varying number of sensitive rules	36
Table 7: Performance Evaluation by varying Support and Confidence.....	36

CHAPTER 1. INTRODUCTION

Data mining an interdisciplinary subfield of computer science is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. It is the process of extracting useful information or knowledge from large databases. Data mining was developed as an important technology for large databases. Data mining applications are business, marketing, medical analysis, products control, quality improvement and scientific research etc.

1.1 APPLICATIONS

CRIME AND ANTI-TERRORISM AGENCIES ^[1]

Total Information Awareness (TIA) is the name of a massive U.S. data mining project focused on scanning travel, financial and other data from public and private sources with the goal of detecting and preventing transnational threats to national security. TIA has also been called Terrorism Information Awareness. The program was part of the Homeland Security Act and, after its creation in January 2003, was managed by the Defence Advanced Research Projects Agency (DARPA). The basic idea was to gather as much data as possible about everyone, sift through it with massive computers, and investigate patterns that might indicate terrorist plots.

SERVICE PROVIDERS

The example of Data Mining and Business Intelligence comes from service providers in the mobile phone and utilities industries. Mobile phone and utilities companies use

Data Mining and Business Intelligence to predict ‘churn’, the terms they use for when a customer leaves their company to get their phone/gas/broadband from another provider. They collate billing information, customer services interactions, website visits and other metrics to give each customer a probability score, then target offers and incentives to customers whom they perceive to be at a higher risk of churning.

E-COMMERCE

Perhaps some of the most well-known examples of Data Mining and Analytics come from E-commerce sites. Many E-commerce companies use Data Mining and Business Intelligence to offer cross-sells and up-sells through their websites. One of the most famous of these is, of course, **Amazon**, that uses sophisticated mining techniques to drive their 'People who viewed that product, also liked this' functionality.

1.2 MOTIVATION

Data mining has operated on a data-warehousing model of gathering all data into a central site, then running an algorithm against that data. Privacy considerations may prevent this approach. For example, the Centres for Disease Control may want to use data mining to identify trends and patterns in disease outbreaks, such as understanding and predicting the progression of a flu epidemic. Insurance companies have considerable data that would be useful – but are unwilling to disclose this due to patient privacy concerns. An alternative is to have each of the insurance companies provide some sort of statistics on their data that cannot be traced to individual patients, but can be used to identify the trends and patterns of interest to the Centre for Disease Control.

Privacy-preserving data mining has emerged to address this issue. One approach is to alter the data before delivering it to the data miner. The second approach assumes the data is distributed between two or more sites, and these sites cooperate to learn the global data mining results without revealing the data at their individual sites. This approach was first introduced to the data mining community, with a method that enabled two parties to build a decision tree without either party learning anything about the other party's data, except what might be revealed through the final decision tree.

1.3 OBJECTIVE

Objective is to provide privacy preservation in data mining. The main concern for an organisation participating data mining is preserving privacy of the data of the organisation.

For example, A medical organisation provides data to a third party for mining. The data includes age of patient, pin code of his address and disease (which is a secret information

and is to be mined). The name of the patient is not provided for the sake confidentiality and secrecy of the patient. But an attacker might obtain this information i.e. the name of the patient from an already existing public database like the Voter's list. The attacker can then extract the required information from the public database i.e. Voter's list by matching the pin code and age of the patient in the data set available for mining.

This poses a threat to the privacy for an organisation. I will, therefore, devise and implement an algorithm for data hiding to preserve this privacy by storing data on a cloud.

1.4 CONTRIBUTION

I have until now simulated and evaluated results of the existing algorithm for Association Rule Mining called the Apriori algorithm. I have studied the various Data Hiding techniques and compared them to find the strengths and drawbacks of each technique. I have also implemented one of the data hiding techniques by combining it with the Apriori algorithm to achieve the motive of privacy preservation in distributed data mining.

1.5 ORGANISATION OF PROJECT REPORT

The report follows from general discussion on Data Mining and its applications, followed by explanation of association rule mining. In chapter 2 I detail the theoretical background along with the literature survey. Chapter 3 further includes the Apriori algorithm for association rule mining and a discussion on data hiding techniques that are used for preserving privacy in data mining. I have also mentioned the performed simulation results for Apriori algorithm and this data hiding technique in chapter 4. What I intend to fulfil in this project and the work done till the second phase of the project is also stated.

CHAPTER 2. THEORETICAL BACKGROUND AND LITERATURE SURVEY

I have identified two broad implementation areas of Privacy preservation in data mining namely,

1. Secure Multiparty Computation (SMC)
2. Data Hiding

Here I will discuss the above two classified areas and the feasibility of implementation of the two methods. I will then elaborate my discussion on data hiding on a cloud setup and provide a theoretical background for this approach. I will also examine the prerequisite required, for implementing the above data hiding method, namely Association Rule Mining.

2.1 ASSOCIATION RULE MINING

Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. eg if a customer buys a bread packet, he is 80% likely to also purchase butter.

2.1.1 NEED FOR ASSOCIATION RULE MINING [2]

Association rules are a substantial part of every e-shop, of every supermarket and every tool that aims to analyse data. When I buy something at amazon, I always notice that they are kind of obsessed with showing the items related to my order. Where do they get this information? It is not stored statically in the database, instead it is computed from the overall orders using the association rules mining algorithms.

Customers Who Bought This Item Also Bought



Items are organized in a way that maximizes a chance that the items are bought. Again, this is information that can be easily discovered using association rules mining algorithms

Association rule is an implication of form $A \rightarrow B$, where the left side, A, is called premise and it represents a condition which must be true, for the right side, B (conclusion) to hold. A rule $A \rightarrow B$ can be interpreted as “If A happens, then B happens.”

Bread \rightarrow Milk: Customers, who bought bread, also bought milk

This is done so that they can act based on the knowledge. One can move the milk closer to bread to sell more of it together and generate more income.

So to get started for deriving such rules one need Dataset in the transaction form .The transaction is a logical group of somehow related items. Dataset might have groups of market basket items, groups of links clicked on one web page visit, group of one patient's diseases. Such groups are then called transactions.

2.1.2 EXPLANATION WITH EXAMPLE

Id	Transactions	bread	bread - cheese	
		+ cheese	> cheese	-> bread
1	bread, cheese, honey, apples	✓	✓	✓
2	milk, bread, cheese, pasta	✓	✓	✓
3	milk, bread, apples		✗	
4	bread, milk		✗	
5	milk, pasta, cheese			✗
6	milk, bread, cheese	✓	✓	✓

Table 1. Example of ARM

Bread -> cheese

This rule is found in transactions 1,2,6. Although association rule mining may seem like a very trivial task at the first look, imagine finding the rules in dataset of billions of transactions.

There is no way to tell which rule is better, it is impossible to compare them. To get past this limitation, I can add several classifiers to the rule, which will represent the strength of the rule. They are commonly known as interestingness measures, because the strength of the rule is equal to its interestingness. The two classical measures are:

1. **Support** is a measure, which represents how often the rule was applied. It is a percentage of all transaction, where the items in the rule were found.
2. **Confidence** is a percentage of all transactions, which contain items on the left and on the right side of the rule.

No. of transaction in Bread and cheese can be found in transactions: 3 (1, 2, 6)

Total transactions: 6

Support of the rule bread -> cheese and cheese -> bread: 3/6 or 50%.

Now take the rule bread -> cheese.

Our customers bought bread in transactions 1, 2, 3, 4, 6, but bought cheese only in transactions 1, 2 and 6. So five customers bought bread, but only three of them bought also a cheese, so the confidence of the rule is 3/5.

I have implemented data hiding approach over association rule mining concept using Apriori Algorithm and discussed further on data hiding on a cloud set up. But before I proceed with this discussion, I have given a brief description of the secure multiparty computation and its feasibility.

2.2 SECURE MULTIPARTY COMPUTATION [1]

A set of parties with private inputs wish to compute some joint function of their inputs. Parties wish to preserve some security properties. E.g., privacy and correctness. Security must be preserved in the face of adversarial behaviour by some of the participants, or by an external party. It is a mechanism to provide collaborate computations of multiple organizations without revealing data of individual organization.

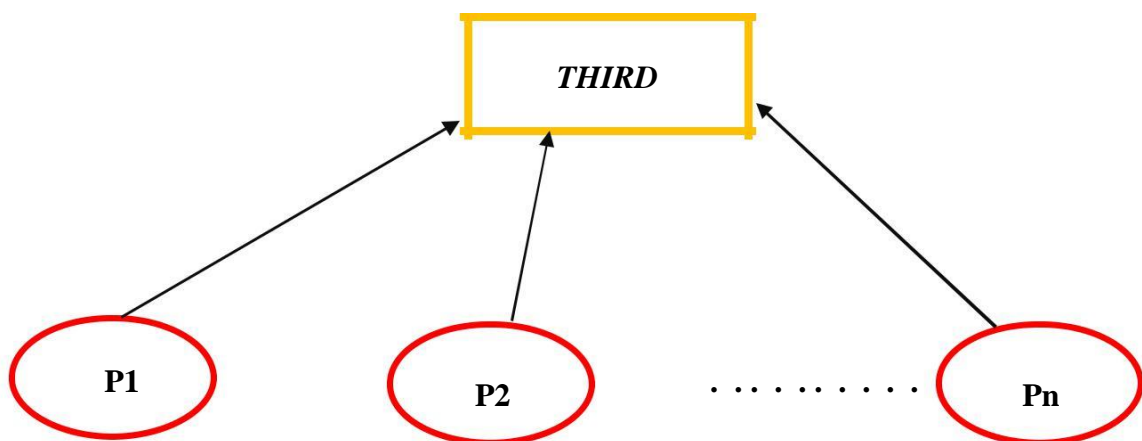


Figure 1. Structure of SMC

Let several organizations P_1, \dots, P_n wish to perform a joint computation. According to SMC, such a computation should be carried out in such a manner that no organization can know the input from other organizations. Thus, SMC is a technique for Privacy Preserving Data Mining in which several parties collaborate perform a joint computation and each party only gets the

final results of computation without knowing the inputs from other parties. Each organization knows nothing except the final computation results.

The basic idea of Secure Multiparty Computation is that a computation is secure if at the end of the computation, no party knows anything except its own input and the results. One way to view this is to imagine a trusted third party – everyone gives their input to the trusted party, who performs the computation and sends the results to the participants. Now imagine that I can achieve the same result without having a trusted party. Obviously, some communication between the parties is required for any interesting computation – how do I ensure that this communication doesn't disclose anything? The answer is to allow non-determinism in the exact values sent in the intermediate communication (e.g., encrypt with a randomly chosen key), and demonstrate that a party with just its own input and the result can generate a “predicted” intermediate computation that is as likely as the actual values. However the general method given does not scale well to data mining sized problems.

2.2.1 FEASIBILITY OF SMC

For mutual benefit, organizations tend to share their data for analytical purposes, thus raising privacy concerns for the users. Over the years, numerous attempts have been made to introduce privacy and security at the expense of massive additional communication costs. The approach of protocols such as Secure Multiparty Computation (SMC) suggested in the literature has proven communication overheads. And in practice are found to be slower by a factor of more than 106. In light of the practical limitations posed by privacy using the traditional approaches, I explore a paradigm shift to side-step the expensive protocols of SMC. In this work, I use the paradigm of data hiding, which allows the data to be divided into multiple shares and processed separately at different servers. Using the paradigm of data hiding, allows me to design a provably-secure, cloud computing based solution which has negligible communication overhead compared to SMC and is hence over a million times faster than similar SMC based protocols [2].

2.3 DATA HIDING

Data mining techniques have been widely used in various applications. However, the misuse of these techniques may lead to the disclosure of sensitive information. Researchers have recently made efforts at hiding sensitive association rules. Nevertheless, undesired side effects, e.g., non-sensitive rules falsely hidden and spurious rules falsely generated, may be produced in the rule hiding process.

Through this project, I present a novel approach that strategically modifies a few transactions in the transaction database to decrease the supports or confidences of sensitive rules without producing the side effects. Since the correlation among rules can make it impossible to achieve this goal, in this paper, I propose heuristic methods for increasing the number of hidden sensitive rules and reducing the number of modified entries. The experimental results show the effectiveness of this approach, i.e., undesired side effects are avoided in the rule hiding process. The results also report that in most cases, all the sensitive rules are hidden without spurious rules falsely generated. Moreover, the good scalability of this approach in terms of database size and the influence of the correlation among rules on rule hiding are observed.

2.4 HIDING TECHNIQUES

Given a database D , a set R of relevant rules that are mined from D and a subset RH (sensitive rules) of R , I have to transform D into a database D' in such a way that the rules in R can still be mined. Two main approaches for implementing the above are:

1. I can either prevent the rules in RH from being generated by hiding the frequent itemsets from which they are derived.
2. I can reduce the confidence of the sensitive rules by bringing it below a user-specified threshold (min_conf).

2.4.1 ACCURACY OF HIDING ALGORITHMS

On changing the raw database, there will be many negative impacts that can be classified into two parts:

1. Useful rules have been lost.
2. New rules have been produced artificially.

Accuracy of the hiding technique will depend on how it hides all sensitive rules in less time complexity along with reducing these negative impacts.

2.4.2 THEORETICAL COMPARISON OF HIDING TECHNIQUES

Three algorithms have been studied and compared. The algorithms for these algorithms have been discussed in the next chapter. Here I have shown a comparative study of the three rules hiding algorithms namely ISLF, DSRF and MDSRRC.

Advantages of DSRF

- In this algorithm, I are deleting items that are present in consequents of sensitive rules, from the transactions that support this sensitive rule.
- Thus, support of consequents decreases and in turn confidence of the rule decreases.
- So **no false rule generation**.

Disadvantages of DSRF

- Takes more time to compute.
- Make more modification in the database as item deletion procedure is performed for every sensitive rule.

Disadvantages of ISLF

- In this algorithm, I are adding items that are present in antecedents of sensitive rules, in the transactions that does not support these sensitive rule.
- Thus, support of antecedent increases and in turn confidence of the rule decreases.
- However, in doing so, many different and useless item sets will be generated.

- This will lead to **false rule generation**.
- Moreover, there will be chances that same antecedents are also present in some useful rules. Thus, **useful rules will also be lost**.

Advantages of MDSRRC

- Sensitive rules are hidden more efficiently.
- No false rule generation
- Sensitive rules are decided by the user/database owner instead of deciding it on the basis of support and confidence.
- Less time complexity.
- Less modification done in database as deletion is performed after analysing all sensitive rules.

CHAPTER 3. PROPOSED MODEL AND WORK

In this chapter, I have discussed Apriori Algorithm for Association Rule Mining and three Data Hiding techniques.

Apriori Algorithm [3] is used to generate association rules for a given data set efficiently by pruning the irrelevant sets. The Data Hiding techniques are used to modify the database so that the sensitive rules cannot be mined and hence privacy is preserved.

3.1 APPROACH TO ARM

Two-step approach:

1. Frequent Item set Generation – Generate all item sets whose support \geq minsup (Threshold support)
2. Rule Generation – Generate high confidence rules from each frequent item set, where each rule is a binary partitioning of a frequent item set.

3.1.1 FREQUENT ITEMSET GENERATION

Suppose there are 'd' number of items.

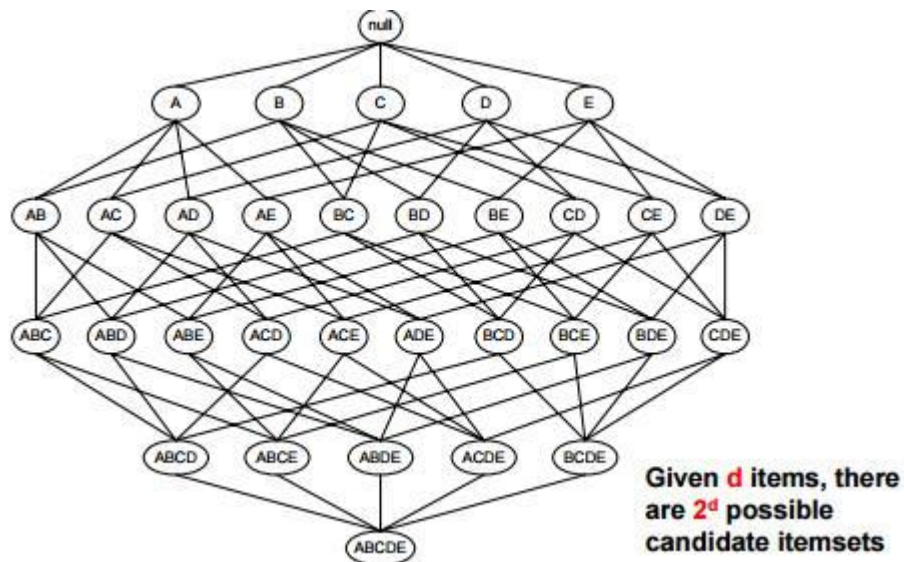


Figure 2. Frequent Item Set generation in ARM

Brute-force method (for small item sets):

Generate all possible subsets of an item sets, excluding the empty set ($2^d - 1$) and use them as rule consequents (the remaining items form the antecedents).

Compute the confidence: divide the support of the item set by the support of the antecedent (get it from the hash table).

Select rules with high confidence (using a threshold). So for Given d unique items:

- Total number of item sets = 2^d
- Total number of possible association rules[3]:

$$R = \sum_{k=1}^{d-1} \left[\binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j} \right] \quad \dots (1)$$
$$= 3^d - 2^{d+1} + 1$$

If $d = 6$, $R = 602$ rules

It is computationally expensive especially when I are dealing with large data items. Therefore, Apriori algorithm is used as pruning technique to reduce total number of item sets.

APRIORI PRINCIPLE [3]

If an item set is frequent, then all of its subsets must also be frequent, or if an item set is infrequent then all its supersets must also be infrequent

Apriori principle holds due to the following property of the support measure:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

Support of an item set never exceeds the support of its subsets.

This is known as the **anti-monotone** property of support

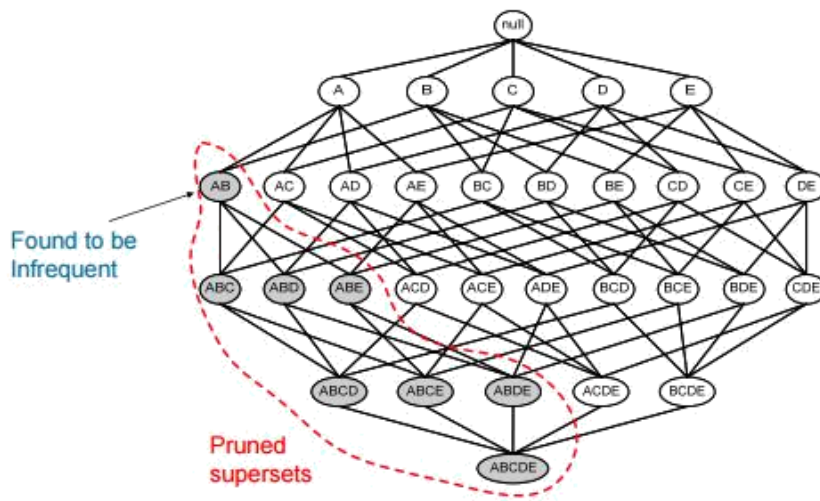


Figure 3. Pruned Structure of ARM

Hence here as item set $\{A, B\}$ is infrequent its further level sets will also be infrequent. Thus I can prune its subtree, hence reducing many useless cases.

Level-wise algorithm:

1. Let $k = 1$
2. Generate frequent item sets of length 1
3. Repeat until no new frequent item sets are identified
 1. Generate length $(k+1)$ candidate item sets from length k frequent item sets
 2. Prune candidate item sets containing subsets of length k that are infrequent
 3. Count the support of each candidate by scanning the DB
 4. Eliminate candidates that are infrequent, leaving only those that are frequent

Note: steps 3.2 and 3.4 prune item sets that are infrequent

3.1.2 RULE GENERATION

Given a frequent item set L , find all non-empty subsets $f \subset L$ such that $f \rightarrow L - f$ satisfies the minimum confidence requirement

If $\{A, B, C, D\}$ is a frequent item set, candidate rules:

- ABC \rightarrow D,
- ABD \rightarrow C,
- ACD \rightarrow B,
- BCD \rightarrow A,
- A \rightarrow BCD,
- B \rightarrow ACD
- C \rightarrow ABD,

$D \rightarrow ABC$ $AB \rightarrow CD$,
 $AC \rightarrow BD$,
 $AD \rightarrow BC$,
 $BC \rightarrow AD$,
 $BD \rightarrow AC$,
 $CD \rightarrow AB$,

If $|L| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \emptyset$ and $\emptyset \rightarrow L$)

Efficiently generate rules from frequent item sets:

In general, confidence does not have an anti-monotone property $c(ABC \rightarrow D)$ can be larger or smaller than $c(AB \rightarrow D)$ – But confidence of rules generated from the same item set have an anti-monotone property

e.g., $L = \{A, B, C, D\}$: $c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$

Confidence is anti-monotone w.r.t. number of items on the RHS of the rule.

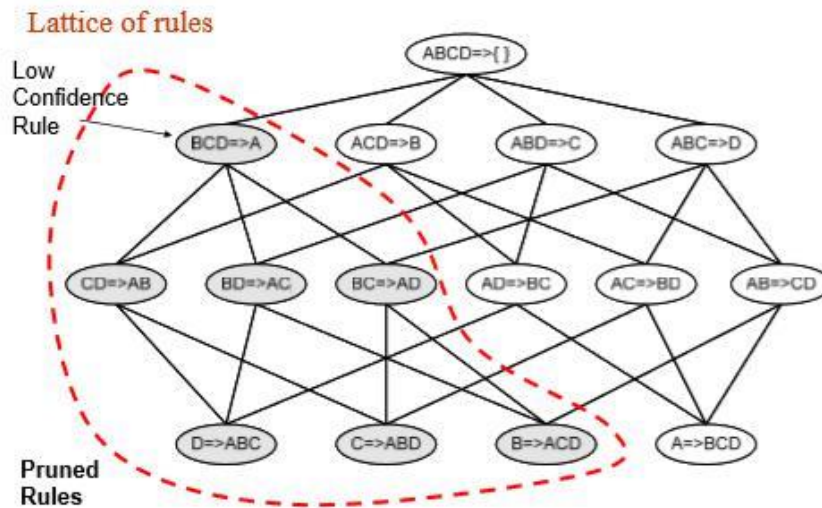
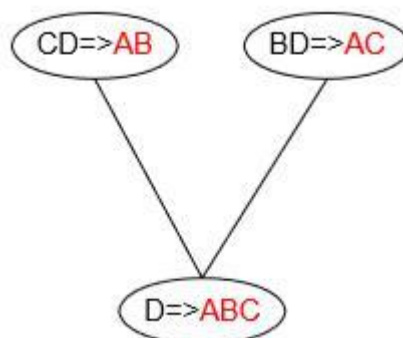


Figure 4. Pruned Structure of Lattice of rules of ARM

Candidate rule is generated by merging two rules that share the same prefix in the rule consequent.



Join($CD \Rightarrow AB$, $BD \Rightarrow AC$) would produce the candidate rule $D \Rightarrow ABC$

Prune rule $D \Rightarrow ABC$ if does not have high confidence

Support counts have been obtained during the frequent item set generation step

3.2 DATA HIDING TECHNIQUES

3.2.1 PROBLEM FORMULATION

Association rules using support and confidence can be defined as follows.

Let $I = \{I_1, I_2 \dots I_m\}$ be a set of items.

Let $D = \{T_1, T_2 \dots T_n\}$ be a set of transactions. where each transaction T in D is a set of items such that $T \subseteq I$ an association rules of implication in the form of $X \rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \emptyset$.

(a) Sample Database D and (b) Large Itemsets Obtained from D

TID	Items
T1	ABC
T2	ABC
T3	ABC
T4	AB
T5	A
T6	AC

Table 2.

Itemset	Support
A	100%
B	66%
C	66%
AB	66%
AC	66%
BC	50%
ABC	50%

Table 3.

The Rules Derived from the Large Itemsets of Table 3

Rules	Confidence	Support
$B \Rightarrow A$	100%	66%
$B \Rightarrow C$	75%	50%
$C \Rightarrow A$	100%	66%
$C \Rightarrow B$	75%	50%
$B \Rightarrow AC$	75%	50%
$C \Rightarrow AB$	75%	50%
$AB \Rightarrow C$	75%	50%
$AC \Rightarrow B$	75%	50%
$BC \Rightarrow A$	100%	50%

Table 4.

The Sample Database that Uses the Proposed Notation

TID	Items	Size
T1	111	3
T2	111	3
T3	111	3
T4	110	2
T5	100	1
T6	101	2

Table 5.

Association rule mining

1. Support: $XUY (XUY / |D|) \geq MST$
2. Confidence: $X \rightarrow Y (XUY/X) \geq MCT$

A class of modification. Given two transaction sets Σ_1 and Σ_2 , a class of modification is a function $\phi: (\Sigma_1, I, O) \rightarrow \Sigma_2$ that transforms Σ_1 to Σ_2 , where I is the item(s) to be modified and O is the modification scheme.

Association rule hiding. Let D' be the database after applying a sequence of modification to D .

A strong rule $X \rightarrow Y$ in D will be hidden in D' if one of the following condition holds in D' .

1. Support: $XUY < MST$
2. Confidence: $X \rightarrow Y < MCT$

3.2.2 RULE HIDING ALGORITHMS

I propose two data mining algorithms for hiding sensitive association rules, namely Increase Support of LHS (ISLF) [4] and Decrease Support of RHS (DSRF) [4].

The first algorithm tries to increase the support of left hand side of the rule.

The second algorithm tries to decrease the support of the right hand side of the rule. The details of the two algorithms are described as follow.

1. ALGORITHM (ISLF) [4]

Input:

1. A source database D ,
2. A min_ support,
3. A min_ confidence,
4. A set of predicting items X

Output:

A transformed database D' , where rules containing X on LHS will be hidden

```

Find large 1-itemsets from D;
for each predicting item  $x \in X$ 
    If  $x$  is not a large 1-itemset, then  $X := X - \{x\}$ ;
    If  $X$  is empty, then EXIT;
Find large 2-itemsets from D;
    for each  $x \in X$  {
        for each large 2-itemset containing  $x$  {
            Compute confidence of rule  $U$ , where  $U$  is a rule like  $x \rightarrow y$ ;
            if  $\text{conf}(U) < \text{min\_conf}$ , then
                Go to next large 2-itemset;
            else { //Increase Support of LHS
                //  $T$  is the transaction list
                Find  $T = \{t \text{ in } D / t \text{ does not support } U\}$ ;
            };
            Sort  $TL$  in ascending order by the number of items;
            While  $\text{conf}(U) \geq \text{min\_conf}$  and  $TL$  is not empty {
                Choose the first transaction  $t$  from  $TL$ 
                Modify  $t$  to support  $x$ , the LHS ( $U$ );
                Compute support and confidence of  $U$ ;
                Remove and save the first transaction  $t$  from  $TL$ ;
            }; // end While
        }; // end if  $\text{conf}(U) < \text{min\_conf}$ 

        If  $TL$  is empty, then {
            Cannot hide  $x \rightarrow y$ ;
            Restore  $D$ ;
            Go to next large-2 item set;
        } // end if  $TL$  is empty

    } // end of for each large 2-itemset

Remove  $x$  from  $X$ ;

```

```
} // end of for each  $x \in X$ 
```

Output updated D, as the transformed D':

2. ALGORITHM (DSRF) [4]

Input:

1. A source database D,
2. A min_ support,
3. A min_ confidence,
4. A set of predicting items X

Output:

A transformed database D', where rules containing X on LHS will be hidden

Find large 1-item sets from D;

for each predicting item $x \in X$

If x is not a large 1-itemset, then $X := X \setminus \{x\}$;

If X is empty, then EXIT;

Find large 2-itemsets from D;

for each $x \in X$ {

for each large 2-itemset containing x {

Compute confidence of rule U, where U is a rule like $x \rightarrow y$;

If $\text{conf}(U) < \text{min_conf}$, then

Go to next large 2-itemset;

else { //Decrease Support of RHS

Find $TR = \{t \text{ in } D / t \text{ fully support } U$

};

Sort TR in ascending order by the number of items;

While $\{\text{conf}(U) \geq \text{min_conf} \text{ and } TR \text{ is not empty}\}$ {

Choose the first transaction t from TR;

Modify t so that y is not supported;

Compute support and confidence of U;

Remove and save the first transaction t from TR;

}; // end While

```

}; // end if conf(U) < min_conf

If TR is empty, then {
    Cannot hide  $x \rightarrow y$ ;
    Restore D;
    Go to next large-2 item set;
} // end if TR is empty
} // end of for each large 2-itemset
Remove x from X;
} // end of for each  $x \in X$ 

Output updated D, as the transformed D';

```

3. ALGORITHM (MDSRRC) [5]

Input:

1. MCT (Minimum Confidence Threshold),
2. Original database D,
3. MST (Minimum support threshold).

Output:

Database D' with all sensitive rules are hidden.

1. Apply Apriori algorithm [3] on given database D. Generate all possible association rules R.
2. Select set of rules SR in R as sensitive rules.
3. Calculate sensitivity of each item j in D.
4. Calculate sensitivity of each Transaction.
5. Count occurrences of each item in R.H.S of sensitive rules, find $IS = \{is_0, is_1 \dots is_k\} \ k \leq n$, by arranging those items in descending order of their count. If two items have same count then sort those in descending order of their actual support count.
6. Select the transactions which supports is_0 , then sort them in descending order of their sensitivity. If two transactions have same sensitivity then sort those in increasing order of their length.

While (SR is not empty) {

Start with first transaction from sorted transactions,

Delete item is_0 from that transaction.

for each rule r in SR{

Update support and confidence of the rule r.

If (support of r < MST or confidence of r < MCT) {

Delete Rule r from SR.

Update sensitivity of each item.

Update IS (This may change is_0).

Update the sensitivity of each transaction.

Select the transactions, which are supports is_0 ,

Sort those in descending order of their sensitivity.

}

Else {

Take next transaction from sorted transactions, go to step 10.

}

}

}

End

3.3 PROPOSED CLOUD MODEL

I have designed a software as a Service (SaaS) cloud model. It is a web service which provides a user friendly GUI with interactive and easy to use interface features.

The cloud model web service has an HTML frontend for UI and Java backend which runs the various algorithms.

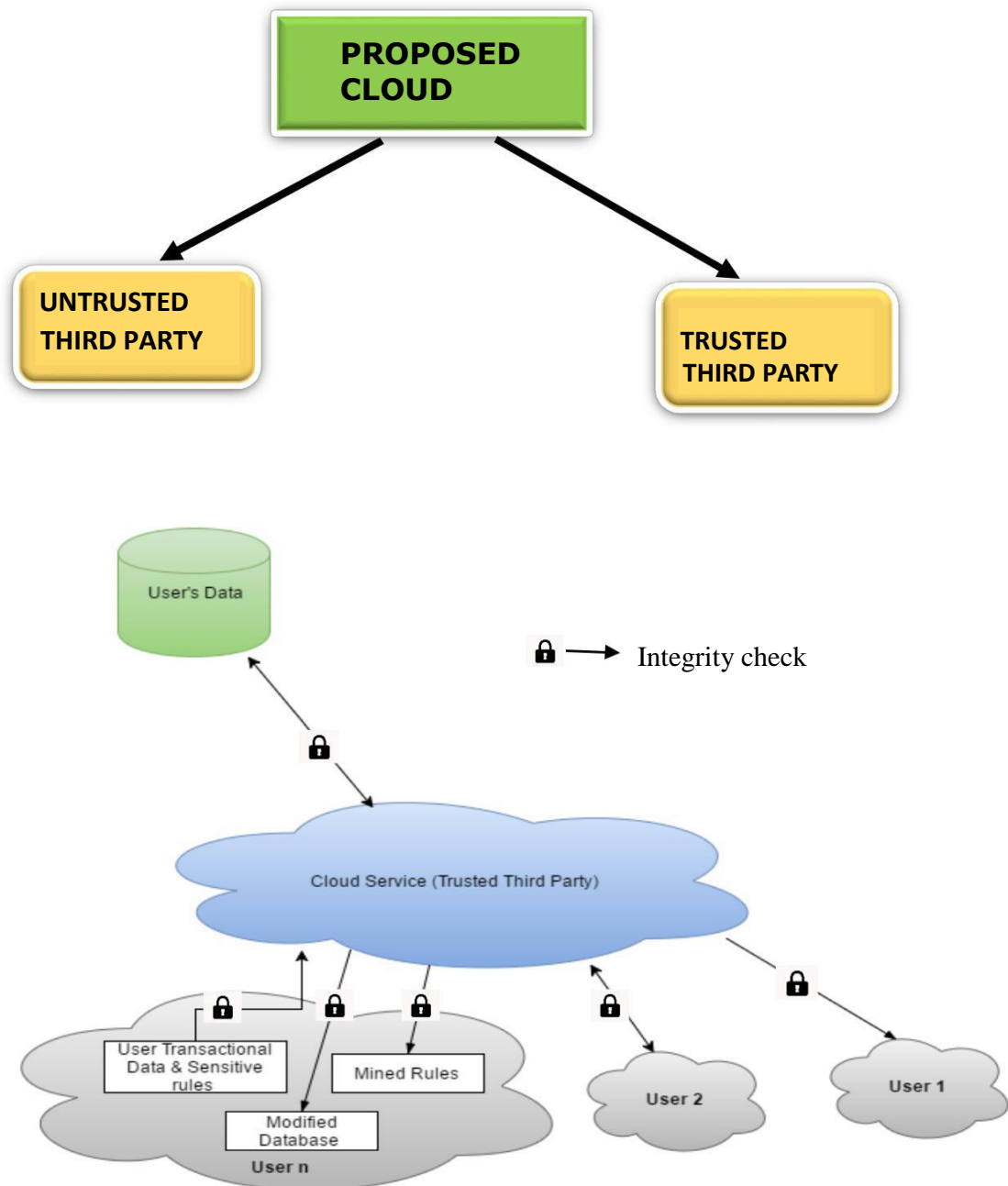


Figure 5. : Proposed Cloud Model

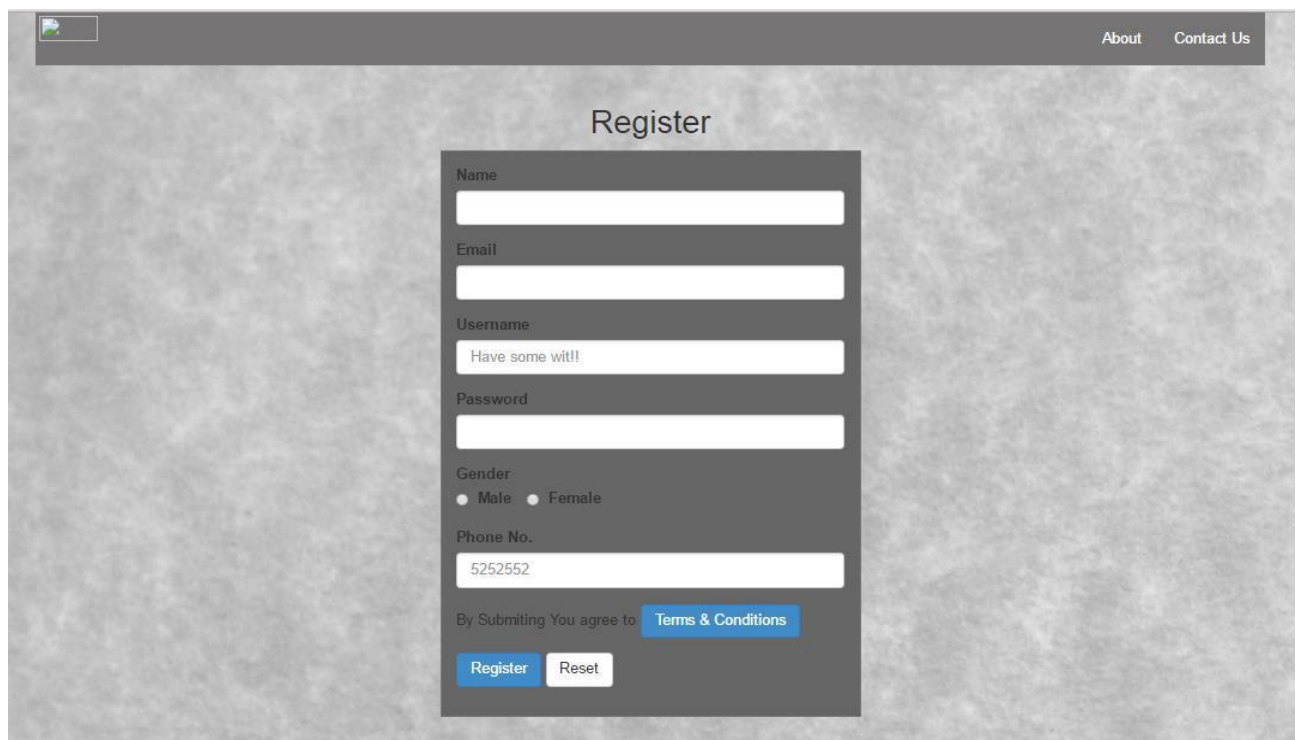
3.3.1 STRENGTHS OF MODEL

- Easy accessibility of the cloud model from anywhere and everywhere
- Hassle-free computation of data to obtain desired results
- Increased availability of resources like storage capacity and computing power
- Large database can be easily stored on the cloud server
- Data integrity is also provided by this model

3.3.2 USER INTERFACE

I have designed a user interface using which the user can perform various operations on the file by applying the rule hiding algorithms.

Shown below is the registration page with which the user can register himself to the cloud services provided by me and create his account to perform data hiding on his database.



The image shows a web registration form titled "Register". The form is centered on a light gray background. At the top right of the page, there are links for "About" and "Contact Us". The form fields are as follows:

- Name:** A text input field.
- Email:** A text input field.
- Username:** A text input field with the placeholder text "Have some wit!!".
- Password:** A text input field.
- Gender:** Radio buttons for "Male" and "Female".
- Phone No.:** A text input field with the placeholder text "5252552".

Below the form fields, there is a line of text: "By Submitting You agree to" followed by a blue button labeled "Terms & Conditions". At the bottom of the form, there are two buttons: a blue "Register" button and a white "Reset" button.

Figure 6.a: Registration Page

With the login page given below the user can log into the account and use his space securely.

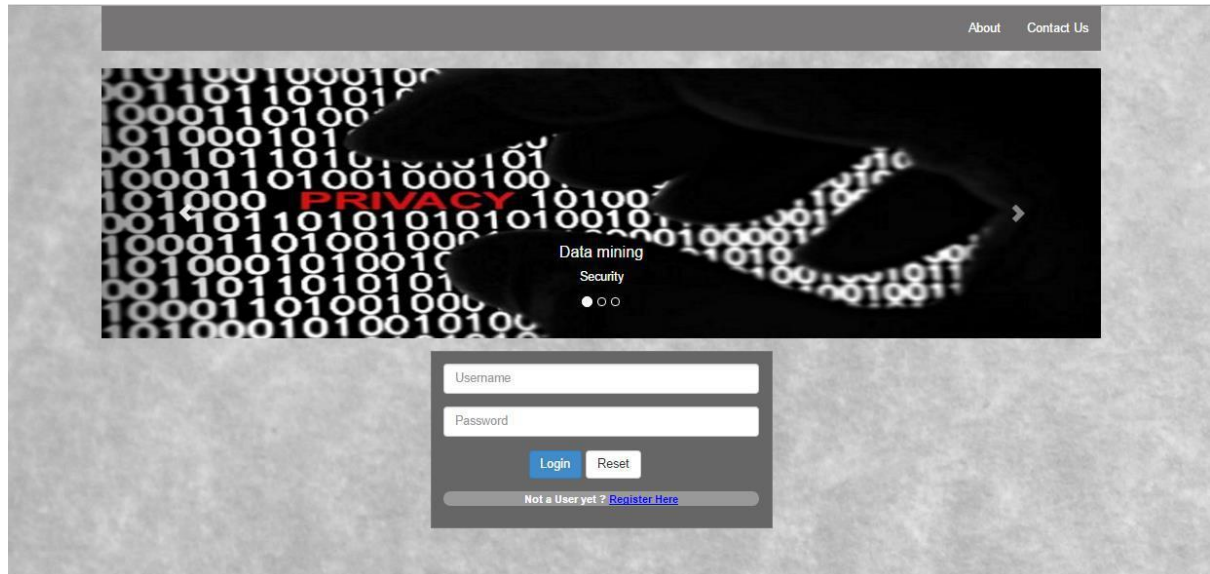


Figure 6.b: Login Page

This is the home page where user has options for performing all file operations, data mining activities and modifying datasets for rule hiding.

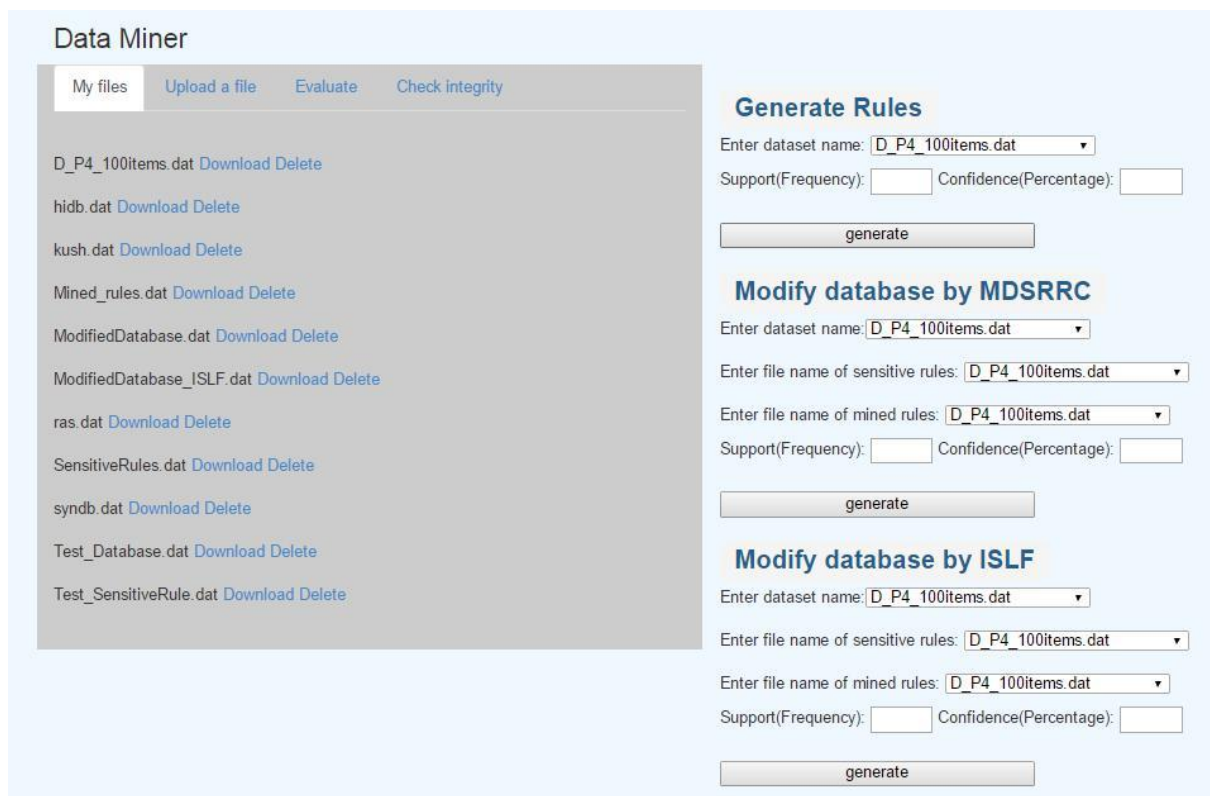
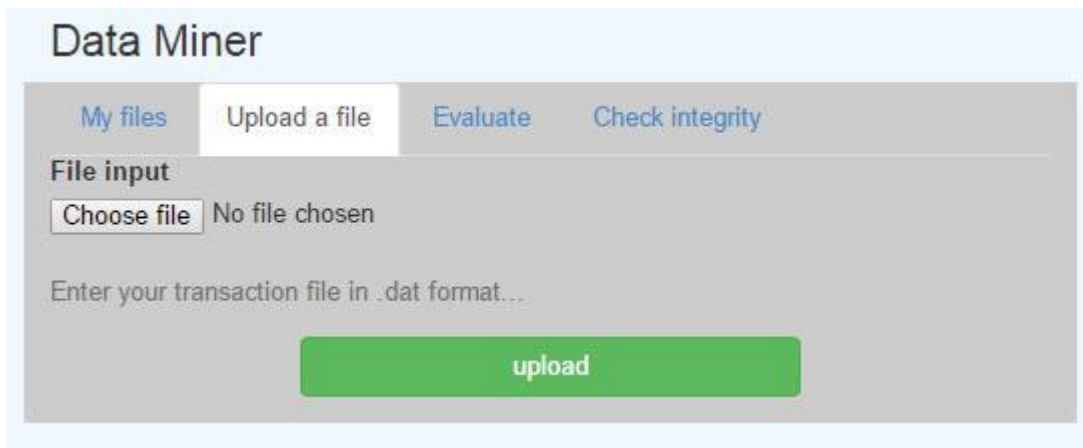


Figure 6.c: User Page

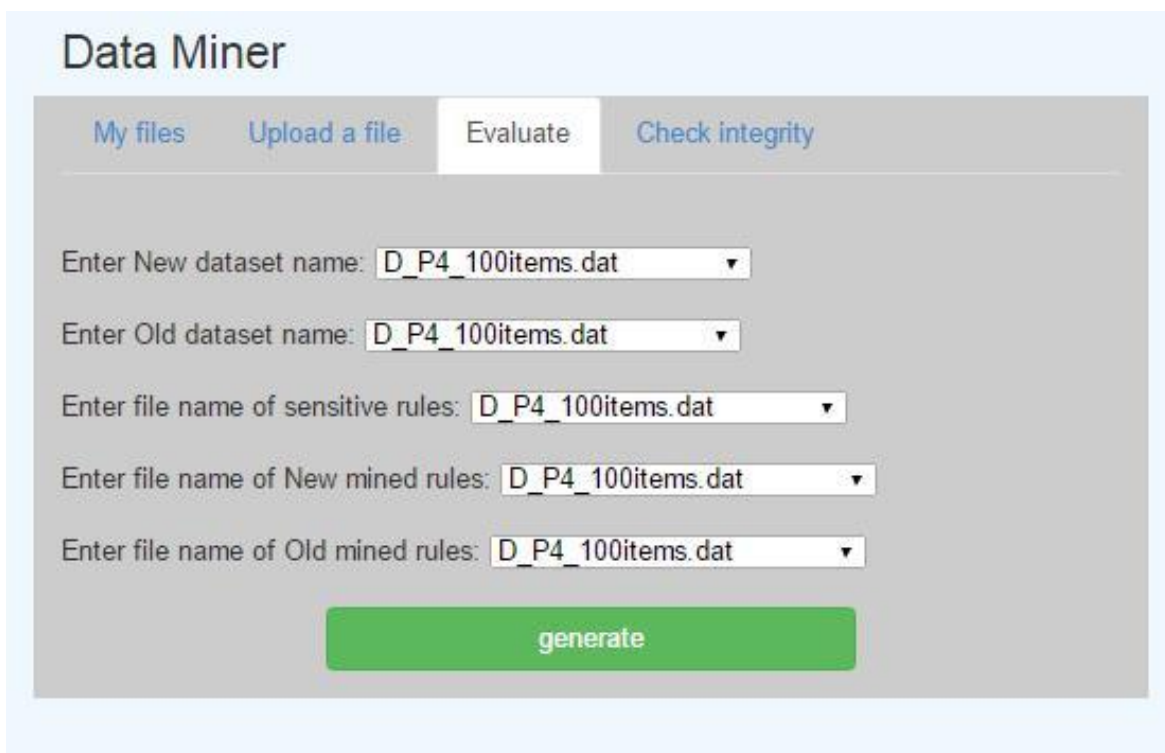
By this option user can upload file.



The screenshot shows the 'Data Miner' interface with the 'Upload a file' tab selected. Under the 'File input' section, there is a 'Choose file' button and the text 'No file chosen'. Below this, it says 'Enter your transaction file in .dat format...'. At the bottom, there is a large green 'upload' button.

Figure 6.d: File upload window

User is also provided with the feature for evaluating the performance of both the algorithms.



The screenshot shows the 'Data Miner' interface with the 'Evaluate' tab selected. It contains five dropdown menus for dataset and rule names, all currently set to 'D_P4_100items.dat'. The labels for these dropdowns are: 'Enter New dataset name:', 'Enter Old dataset name:', 'Enter file name of sensitive rules:', 'Enter file name of New mined rules:', and 'Enter file name of Old mined rules:'. At the bottom, there is a large green 'generate' button.

Figure 6.e: Algorithm Evaluation window

3.3.3 DATA INTEGRITY

Data integrity is a fundamental component of information security.

It is the accuracy and consistency of stored data, indicated by an absence of any alteration in data between two updates of a data record.

This cloud model provides integrity, which secures the data from adversaries. There are two types of adversaries i.e. internal adversary and External Adversary. Internal adversary in this case is the cloud administration and external /outsider adversary can be a network attacker.

I have implemented data integrity using Java's inbuilt function of MD5. The MD5 algorithm hashes the entire file content. The user can store this hash value so that the next time he opens his file he can compare the current generated hash value with the previous value which is stored with him. If the hash values do not match it means that the file was modified by some outsider. Along with hashing the file data, the algorithm also displays the last accessed time which lets user detect any unwanted access or attack on his data. The user can, therefore, check the integrity of his data this way.

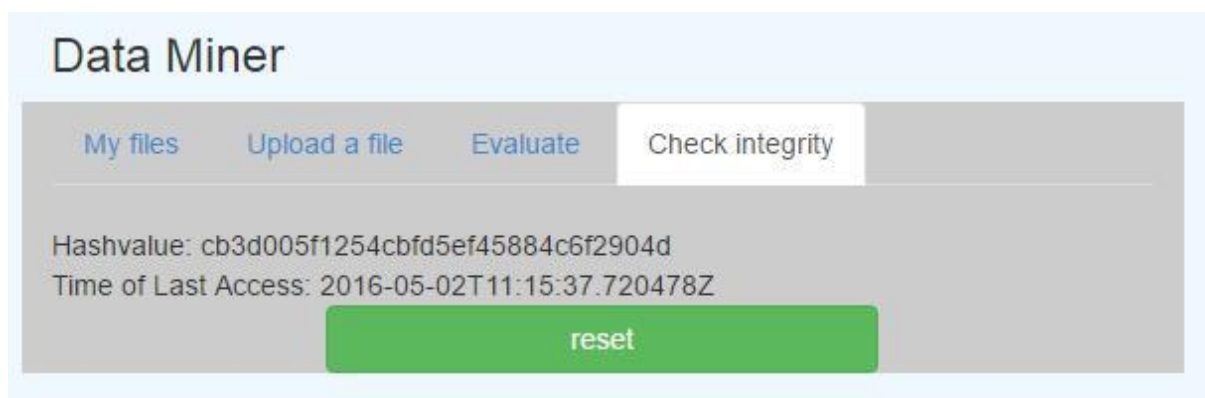


Figure 6.f: Integrity check window

3.4 SUMMARY

The purpose of the Association rule hiding algorithm for privacy preserving data mining is to hide certain crucial information so they cannot be discovered through association rule.

I have proposed an efficient Association rule hiding algorithm for privacy preserving data mining.

This is based on association rule hiding approach of previous algorithms and modifying the database transactions so that the confidence of the association rule can be reduce. In this proposed algorithm I can hide the rules as per user requirement. The user selects certain rules from the mined rules set generated from Apriori and the data hiding algorithm hides those rules by reducing the RHS, so it reduces the number of modification and hides more rules in less time.

The efficiency of the proposed algorithm is compared with ISLF and DSRF approach. This algorithm prunes more number of hidden rules with same number of transactions and modification.

CHAPTER 4. SIMULATION AND RESULTS

I have simulated the Apriori Algorithm and the MDSRRC hiding algorithm.

4.1 HIDING ALGORITHM ALONG WITH APRIORI

The hiding algorithms are implemented using a dataset which covers all the aspects i.e. having different support and confidence variations so that rule derivation can be easily studied. The dataset consists of number of transaction which are composed of different product ids that are bought.

Firstly, I have applied Apriori on the given dataset and generated mined rules. The user specifies the sensitive rules he wants to hide. I then apply the hiding techniques on these rules and hide the sensitive rules by decreasing the support and confidence of these rules.

Transaction 0: [a, b, c, d, e]
Transaction 1: [a, c, d,]
Transaction 2: [a, b, d, f, g]
Transaction 3: [b, c, d, e]
Transaction 4: [a, b, d]
Transaction 5: [c, d, e, f, h]
Transaction 6: [a, b, c, g]
Transaction 7: [a, c, d, e]
Transaction 8: [a, c, d, h]

1	a b c d e
2	a c d
3	a b d f g
4	b c d e
5	a b d
6	c d e f h
7	a b c g
8	a c d e
9	a c d h
10	

Figure 7. Dataset

4.2 SIMULATION RESULT

The screenshot of the results of Apriori and the two Rule Hiding Algorithm are shown below.

4.2.1 RESULTS OF APRIORI ALGORITHM

Firstly, I have applied the Apriori algorithm to this dataset. I have kept the threshold of support as pruning level purely depends on the user requirements i.e. up to which level the user needs the support set consisting of given confidence threshold. Initially only single data set is selected and the item having less than threshold is eliminated so that the tree which is iterated having less support is eliminated.

```
items traverses: []
items traverses: [a]
items traverses: [a, c]
items traverses: []
items traverses: [a]
items traverses: [a, b]
items traverses: [a, b, d]
items traverses: [a, b, d, f]
items traverses: []
items traverses: [b]
items traverses: [b, c]
items traverses: [b, c, d]
items traverses: []
items traverses: [a]
items traverses: [a, b]
items traverses: []
items traverses: [c]
items traverses: [c, d]
items traverses: [c, d, e]
items traverses: [c, d, e, f]
items traverses: []
items traverses: [a]
items traverses: [a, b]
items traverses: [a, b, c]
items traverses: []
items traverses: [a]
items traverses: [a, c]
items traverses: [a, c, d]
items traverses: []
items traverses: [a]
items traverses: [a, c]
items traverses: [a, c, d]
freq set: {[a]=7, [b]=5, [c]=7, [d]=8, [e]=4, [f]=2, [g]=2, [h]=2}
Pruning Level: 1
Pruned tree: {[a]=7, [b]=5, [c]=7, [d]=8, [e]=4}
```

Figure 8.a Pruned tree at level 1

```

Generating next level:2
Level 2 with subset count = {[a, b]=1, [a, c]=1, [a, d]=1, [a, e]=1, [b, c]=1, [b, d]=1, [b, e]=1, [c, d]=1, [c, e]=1, [d, e]=1}
Level 2 pruned based on subset-supset check = {[a, b]=1, [a, c]=1, [a, d]=1, [a, e]=1, [b, c]=1, [b, d]=1, [b, e]=1, [c, d]=1, [c, e]=1, [d, e]=1}
temp after=[a, b]
temp after=[a, c]
temp after=[a, d]
temp after=[a, e]
temp after=[b, c]
temp after=[b, d]
temp after=[b, e]
temp after=[c, d]
temp after=[c, e]
temp after=[d, e]
Level 2 before freq pruning = {[a, b]=4, [a, c]=5, [a, d]=6, [a, e]=2, [b, c]=3, [b, d]=4, [b, e]=2, [c, d]=6, [c, e]=4, [d, e]=4}
Pruning Level: 2
Pruned tree:{[a, b]=4, [a, c]=5, [a, d]=6, [b, c]=3, [b, d]=4, [c, d]=6, [c, e]=4, [d, e]=4}

```

Figure 8.b Pruned tree at level 2

Here prune tree is obtained by discarding the items having support less than threshold support (i.e. 50)

```

Generating next level:3
Level 3 with subset count = {[a, b, c]=3, [a, b, d]=3, [a, c, d]=3, [a, c, e]=1, [a, d, e]=1, [b, c, d]=3, [b, c, e]=1, [b, d, e]=1, [c, d, e]=3}
Level 3 pruned based on subset-supset check = {[a, b, c]=3, [a, b, d]=3, [a, c, d]=3, [b, c, d]=3, [c, d, e]=3}
temp after=[a, b, c]
temp after=[a, b, d]
temp after=[a, c, d]
temp after=[b, c, d]
temp after=[c, d, e]
Level 3 before freq pruning = {[a, b, c]=2, [a, b, d]=3, [a, c, d]=4, [b, c, d]=2, [c, d, e]=4}
Pruning Level: 3
Pruned tree:{[a, b, d]=3, [a, c, d]=4, [c, d, e]=4}

```

Figure 8.c Pruned tree at level 3


```

Generating next level:4
Level 4 with subset count = {[a, b, c, d]=1, [a, c, d, e]=1}
Level 4 pruned based on subset-supset check = {}
Level 4 before freq pruning = {}
Pruning Level: 4
Pruned tree:{}

```

No more levels possible!!!

Figure 8.d Pruned tree at level 4

```

Associtaion Rules:
Rule :1 =>      [a] -> [b]      Confidence=57.142857142857146% Support=4%
Rule :2 =>      [a] -> [c]      Confidence=71.42857142857143% Support=5%
Rule :3 =>      [a] -> [d]      Confidence=85.71428571428571% Support=6%
Rule :4 =>      [a] -> [b, d]   Confidence=42.857142857142854% Support=3%
Rule :5 =>      [a] -> [c, d]   Confidence=57.142857142857146% Support=4%
Rule :6 =>      [b] -> [a]      Confidence=80.0% Support=4%
Rule :7 =>      [b] -> [c]      Confidence=60.0% Support=3%
Rule :8 =>      [b] -> [d]      Confidence=80.0% Support=4%
Rule :9 =>      [b] -> [a, d]   Confidence=60.0% Support=3%
Rule :10 =>     [c] -> [a]      Confidence=71.42857142857143% Support=5%
Rule :11 =>     [c] -> [b]      Confidence=42.857142857142854% Support=3%
Rule :12 =>     [c] -> [d]      Confidence=85.71428571428571% Support=6%
Rule :13 =>     [c] -> [e]      Confidence=57.142857142857146% Support=4%
Rule :14 =>     [c] -> [a, d]   Confidence=57.142857142857146% Support=4%
Rule :15 =>     [c] -> [d, e]   Confidence=57.142857142857146% Support=4%
Rule :16 =>     [d] -> [a]      Confidence=75.0% Support=6%
Rule :17 =>     [d] -> [b]      Confidence=50.0% Support=4%
Rule :18 =>     [d] -> [c]      Confidence=75.0% Support=6%
Rule :19 =>     [d] -> [e]      Confidence=50.0% Support=4%
Rule :20 =>     [d] -> [a, c]   Confidence=50.0% Support=4%
Rule :21 =>     [d] -> [c, e]   Confidence=50.0% Support=4%
Rule :22 =>     [e] -> [c]      Confidence=100.0% Support=4%
Rule :23 =>     [e] -> [d]      Confidence=100.0% Support=4%
Rule :24 =>     [e] -> [c, d]   Confidence=100.0% Support=4%
Rule :25 =>     [a, b] -> [d]   Confidence=75.0% Support=3%
Rule :26 =>     [a, c] -> [d]   Confidence=80.0% Support=4%
Rule :27 =>     [a, d] -> [b]   Confidence=50.0% Support=3%
Rule :28 =>     [a, d] -> [c]   Confidence=66.66666666666667% Support=4%
Rule :29 =>     [b, d] -> [a]   Confidence=75.0% Support=3%
Rule :30 =>     [c, d] -> [a]   Confidence=66.66666666666667% Support=4%
Rule :31 =>     [c, d] -> [e]   Confidence=66.66666666666667% Support=4%
Rule :32 =>     [c, e] -> [d]   Confidence=100.0% Support=4%
Rule :33 =>     [d, e] -> [c]   Confidence=100.0% Support=4%

```

Figure 9. Result of Apriori Algorithm – Mined Rules

4.2.2 RESULTS OF MDSRRC HIDING ALGORITHM

The user specified the sensitive rules to hide and applies the Hiding algorithm on the sensitive rules and the Database. The mined rules from the Apriori algorithm give the support and confidence of each of these rules which are then used by rule hiding algorithm- MDSRRC to modify/ sanitize the database accordingly.

1	[a]->[b]	4	7
2	[a]->[c]	5	7
3	[a]->[d]	6	7
4	[a]->[b, d]	3	7
5	[a]->[c, d]	4	7
6	[b]->[a]	4	5
7	[b]->[c]	3	5
8	[b]->[d]	4	5
9	[b]->[a, d]	3	5
10	[c]->[a]	5	7
11	[c]->[b]	3	7
12	[c]->[d]	6	7
13	[c]->[e]	4	7
14	[c]->[a, d]	4	7
15	[c]->[d, e]	4	7
16	[d]->[a]	6	8
17	[d]->[b]	4	8
18	[d]->[c]	6	8
19	[d]->[e]	4	8
20	[d]->[a, c]	4	8
21	[d]->[c, e]	4	8
22	[e]->[c]	4	4
23	[e]->[d]	4	4
24	[e]->[c, d]	4	4
25	[a, b]->[d]	3	4
26	[a, c]->[d]	4	5
27	[a, d]->[b]	3	6
28	[a, d]->[c]	4	6
29	[b, d]->[a]	3	4
30	[c, d]->[a]	4	6
31	[c, d]->[e]	4	6
32	[c, e]->[d]	4	4
33	[d, e]->[c]	4	4
34			

Figure 10. Mined Rules used for Hiding

```
[a]->[b, d]
[a]->[c, d]
[d]->[a, c]
```

Figure 11. Sensitive Rules defined by user, which are to be hidden

1	a b c e
2	a d
3	a b d f g
4	b c d e
5	a b d
6	c d e f h
7	a b c g
8	a c d e
9	a c d h
10	

Figure 12. Sanitized Database D'

1	[a]->[b] 4 7
2	[a]->[c] 4 7
3	[a]->[d] 4 7
4	[b]->[a] 4 5
5	[b]->[c] 3 5
6	[b]->[d] 3 5
7	[c]->[a] 4 6
8	[c]->[b] 3 6
9	[c]->[d] 4 6
10	[c]->[e] 3 6
11	[c]->[d, e] 3 6
12	[d]->[a] 4 6
13	[d]->[b] 3 6
14	[d]->[c] 4 6
15	[d]->[e] 3 6
16	[d]->[c, e] 3 6
17	[e]->[c] 3 3
18	[e]->[d] 3 3
19	[e]->[c, d] 3 3
20	[c, d]->[e] 3 4
21	[c, e]->[d] 3 3
22	[d, e]->[c] 3 3
23	

Figure 13. Rules generated by Apriori when applied on the new sanitized database

4.2.3 RESULTS OF ISLF HIDING ALGORITHM

I have implemented another rule hiding algorithm – the ISLF algorithm and drawn a comparison of the two algorithms based on certain parameters.

1	[a]->[b]	4 6
2	[a]->[c]	4 6
3	[a]->[d]	5 6
4	[a]->[b, d]	3 6
5	[a]->[c, d]	3 6
6	[b]->[a]	4 5
7	[b]->[c]	3 5
8	[b]->[d]	4 5
9	[b]->[a, d]	3 5
10	[c]->[a]	4 6
11	[c]->[b]	3 6
12	[c]->[d]	5 6
13	[c]->[e]	3 6
14	[c]->[a, d]	3 6
15	[c]->[d, e]	3 6
16	[d]->[a]	5 7
17	[d]->[b]	4 7
18	[d]->[c]	5 7
19	[d]->[e]	3 7
20	[d]->[a, b]	3 7
21	[d]->[a, c]	3 7
22	[d]->[c, e]	3 7
23	[e]->[c]	3 4
24	[e]->[d]	3 4
25	[e]->[c, d]	3 4
26	[a, b]->[d]	3 4
27	[a, c]->[d]	3 4
28	[a, d]->[b]	3 5
29	[a, d]->[c]	3 5
30	[b, d]->[a]	3 4
31	[c, d]->[a]	3 5
32	[c, d]->[e]	3 5
33	[c, e]->[d]	3 3
34	[d, e]->[c]	3 3
35		

Figure 14 . Rules generated by Apriori when applied on the old database

1	[a]->[b]	4 8
2	[a]->[c]	5 8
3	[a]->[d]	7 8
4	[a]->[c, d]	4 8
5	[b]->[a]	4 5
6	[b]->[c]	3 5
7	[b]->[d]	4 5
8	[b]->[a, d]	3 5
9	[c]->[a]	5 6
10	[c]->[b]	3 6
11	[c]->[d]	5 6
12	[c]->[e]	3 6
13	[c]->[a, d]	4 6
14	[c]->[d, e]	3 6
15	[d]->[a]	7 8
16	[d]->[b]	4 8
17	[d]->[c]	5 8
18	[d]->[e]	4 8
19	[d]->[a, c]	4 8
20	[e]->[a]	3 4
21	[e]->[c]	3 4
22	[e]->[d]	4 4
23	[e]->[a, d]	3 4
24	[e]->[c, d]	3 4
25	[f]->[a]	3 3
26	[f]->[d]	3 3
27	[f]->[a, d]	3 3
28	[g]->[a]	3 3
29	[h]->[a]	3 3
30	[h]->[d]	3 3
31	[h]->[a, d]	3 3
32	[a, b]->[d]	3 4
33	[a, c]->[d]	4 5
34	[a, d]->[b]	3 7
35	[a, d]->[c]	4 7
36	[a, d]->[e]	3 7
37	[a, d]->[f]	3 7
38	[a, d]->[h]	3 7

Figure 16. Rules generated by Apriori when applied on the new sanitized database

1	a b c d e
2	a c d
3	a b d f g
4	b c d e
5	a b d
6	c d e f h a
7	a b c g
8	e f g h d a
9	a c d h

Figure 15. Rules generated by Apriori when applied on the new sanitized database

4.3 PERFORMANCE EVALUATION OF MDSRRC AND ISLF

I have compared and evaluated the two algorithms based on various parameters, which are explained below.

4.3.1 PERFORMANCE EVALUATION PARAMETERS

1. **Hiding Failure (HF)^[10]**: This measure quantifies the percentage of the sensitive patterns that remain disclosed in the sanitized dataset. It is defined as the fraction of the sensitive association rules that appear in the sanitized database divided by the ones that appeared in the original dataset.

$$HF = \frac{|SR(D')|}{|SR(D)|} \quad \dots\dots (2)$$

Where $|SR(D')|$ is number of the sensitive rules discovered in the sanitized dataset D' , $|SR(D)|$ is the number of sensitive rules appearing in the original dataset D . Ideally, the hiding failure should be 0%.

2. **Artificial Pattern (AP)^[10]**: This measure quantifies the Percentage of the discovered patterns that are artificial facts.

$$AP = \frac{|P'| - |P \cap P'|}{|P'|} \quad \dots\dots (3)$$

where P is the set of association rules discovered in the original database D and P' is the set of association rules discovered in D'

3. **Dissimilarity (DISS) [10]**: This measure quantifies the amount by which the database is modified while hiding sensitive association rule.

$$Diss(D, D') = \frac{1}{\sum_{i=1}^n f_D(i)} \times \sum_{i=1}^n [f_{D'}(i) - f_D(i)] \quad \dots\dots (4)$$

Where $f_D(i)$ is the count of each item i in the original database and $f_{D'}(i)$ is the count of each item i in the modified database.

4. **Misses Cost (MC)** ^[10]: It is a measure of the number of useful rules that are preserved after modification of database.

$$MC = \frac{|S'R(D)| - |S'R(D')|}{|S'R(D)|} \dots\dots (5)$$

Where $|S'R(D)|$ is the size of the set of all non-sensitive rules in the original database D and $|S'R(D')|$ is the size of the set of all non-sensitive rules in the sanitized database D'.

4.3.2 RESULTS OF PERFORMANCE EVALUATION

Here I have varied the number of sensitive rules and compared the two algorithms based on various parameters. I have used synthetic dataset generated by TARtools^[11].

ALGO	#SR	SUP	CONF	HF (%)	AP	DISS (Out of 9722)	MC (%)	NEW MINED RULES	OLD MINED RULES
MDSRRC	5	150	75	0	1/511	47	55	511	1141
ISLF	5	150	75	0	2/910	136	20	910	1141
MDSRRC	7	150	75	0	0/248	88	78	248	1141
ISLF	7	150	75	0	3/982	212	13	982	1141
MDSRRC	3	150	75	0	0/528	37	53	528	1141
ISLF	3	150	75	0	2/780	120	31	780	1141

Table 6

In this table I have varied the support and then compared the two algorithms based on various parameters.

	#SR	SUP	CONF	HF (%)	AP	DISS (Out of 9722)	MC (%)	NEW MINED RULES	OLD MINED RULES
MDSRRC	5	200	90	0	1/13	12	62	13	37
ISLF	5	200	90	0	0/31	52	52	31	37
MDSRRC	5	150	80	0	0	85	77	253	1141
ISLF	5	150	80	0	1/728	356	36	728	1141
MDSRRC	5	100	70	0	0	158	40	1007	1704
ISLF	5	100	70	0	4/1688	886	14	1688	1704

Table 7

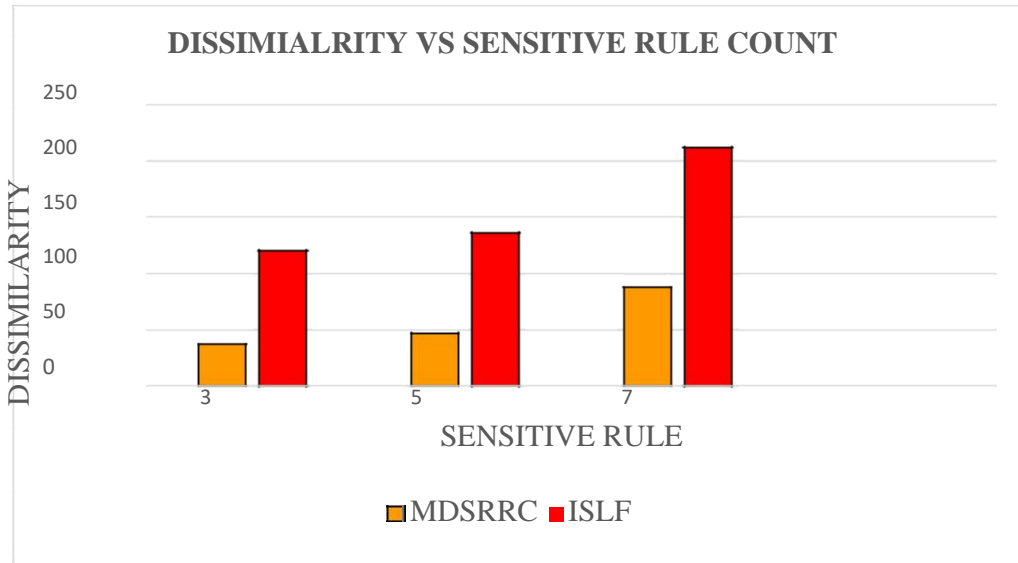


Figure 17. Graph of Dissimilarity vs Sensitive Rule

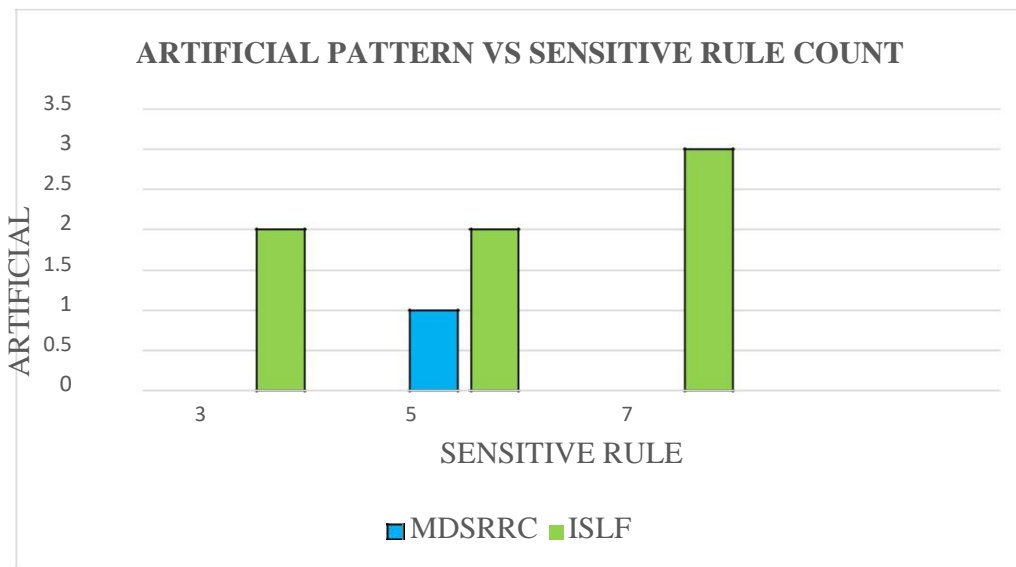


Figure 18. Graph of Artificial Pattern vs Sensitive Rule



Figure 19. Graph of Misses Cost vs Sensitive Rule

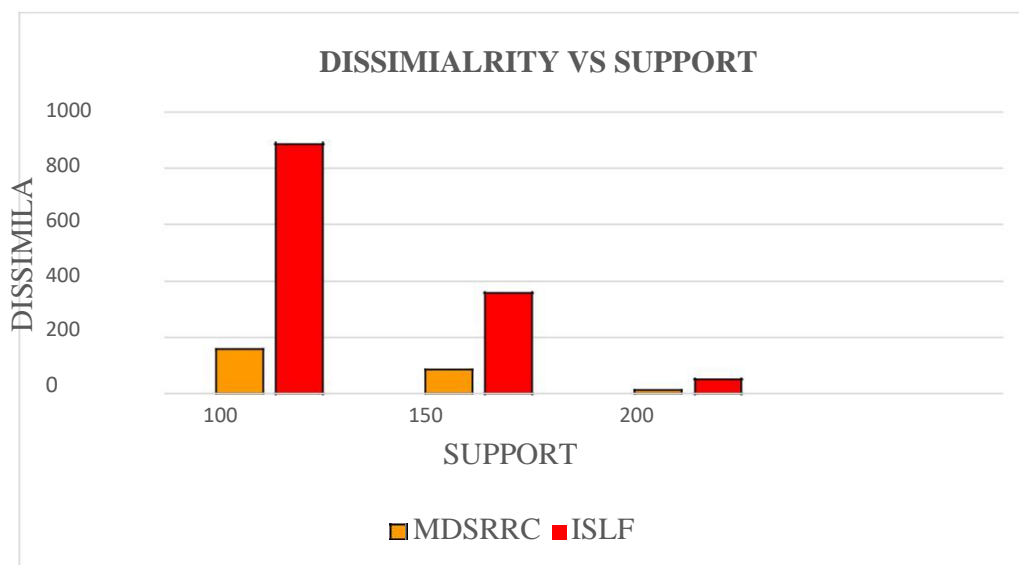


Figure 20. Graph of Dissimilarity vs Support

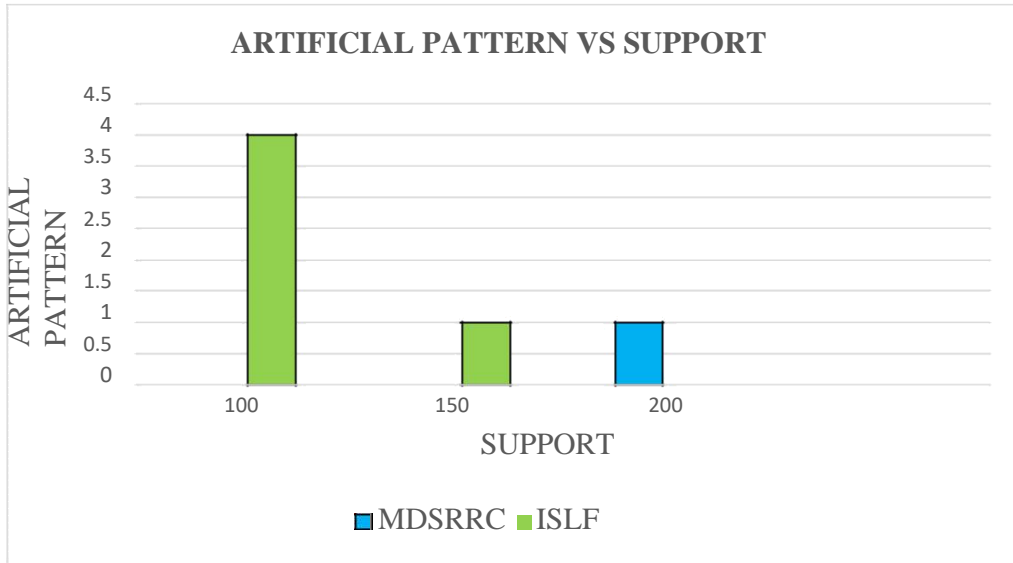


Figure 21. Graph of Artificial Pattern vs Support



Figure 22. Graph of Misses Cost vs Support

CHAPTER 5. CONCLUSION

This model provides a cloud software service that is the whole package of modifying database, publishing it and efficiently mining association rules from it. The model also provides data integrity.

Simulation results proves that MDSRRC can be more efficiently used in hiding knowledge in database as compared to ISLF and DSRF algorithm in terms of dissimilarity and artificial pattern .

As in ISLF, I are adding more items in the transactions, many false rules are being generated. Thus providing incorrect knowledge.

Moreover, ISLF and DSRF are computationally expensive and make more modifications in database.

On other hand, MDSRRC algorithm make minimum modification in database to hide sensitive rules. Its time complexity and number of false rules generated are also less.

REFERENCES

- [1] Techtarget. [Online]. Available: <http://searchsecurity.techtarget.com/definition/Total-Information-Awareness>.
- [2] Maneesh Upmanyu, Anoop M. Namboodiri, Kannan Srinathan, and C.V. Jawahar, "Efficient Privacy Preserving K-Means Clustering," presented at International Institute of Information Technology Hyderabad, 2010.
- [3] [6] Linköping University, [Online]. Available: <http://staffwww.itn.liu.se/~aidvi/courses/06/dm/lectures/lec7.pdf>.
- [4] Nidhi Porwal, Mahavir singh, Sunil Kumar, (2012, Nov.) "An Algorithm for Hiding Association Rules on Data Mining," International Journal of Computer Applications. [Online]. Available: <http://www.ijcaonline.org/proceedings/ctngc/number3/9061-1022> File: ctngc1022.pdf
- [5] Udai Pratap Rao, Nikunj H. Domadiya, "Hiding Sensitive Association Rules to Maintain Privacy and Data Quality in Database," in *Proceedings of IEEE International Advance Computing Conference IACC*, 2013.
- [6] Prof. Hitesh Gupta, Dharmendra Thakur, (2014, Feb.) "Privacy Preservation by Hiding Highly Sensitive Rules with Fewer Side Effects," International Journal of Advanced Research in Computer Science and Software Engineering, pp. 546-551.
- [7] Chen mu yehi, P. C. Yi-Hung Wu, "Hiding Sensitive Association Rules with Limited Side Effects," in *Proceeding of IEEE Transaction on Knowledge and data engineering*, 2007. vol. 19, pp. 29-42.
- [8] M. Gupta. a. R. C. Joshi, (2009, Oct) "Privacy Preserving Fuzzy Association Rules Hiding in Quantitative Data," International Journal of Computer Theory and Engineering, vol. 1, no. 4, pp. 382-388.
- [9] "Association rule mining," [Online]. Available: <http://tkramar.blogspot.in/2008/09/introduction-to-association-rules-minig.html>.
- [10] Atefe, Naderi Dehkordi, Faramarz Safi Esfahani, Ramezani, (2014, June) "Hiding Sensitive Association Rules by Elimination Selective Item among RHS Items for each Selective Transaction," Indian Journal of Science and Technology, vol. 7, no. (6), pp. 831-837.
- [11] Omari, Asem, Regina Langer, and Stefen Conrad "Tartool: A temporal dataset generator for market basket analysis." in *Advanced Data Mining and Applications*, Springer Berlin Heidelberg, 2008, pp. 400-410.