Privacy Preserving Frequent Itemset Mining with Reduced Sensitive Itemsets For Big Data

A

DISSERTATION

Submitted in partial fulfillment of the requirements for the award of degree of

Master of Technology

in

Computer Science and Engineering

Submitted By

HIMANSHU MAKKAR

(16535015)

Department of Computer Science and Engineering

Indian Institute of Technology

Roorkee – 247667

May-2018

CANDIDATE'S DECLARATION

I herby declare that the work presented in this dissertation "***Privacy Preserving Frequent Itemset Mining with Reduced Sensitive Itemsets for Big Data***" towards the fulfillment of the requirements for the award of the degree of ***Master of Technology*** in ***Computer Science and Engineering***, submitted to the Department of Computer Science and Engineering, ***Indian Institute of Technology Roorkee, India*** is an authentic record of my own work carried out during May 2017 to May 2018 under the guidance of ***Dr. Durga Toshniwal, Associate Professor***, Department of Computer Science and Engineering, Indian Institute of Technology, Roorkee.

The content presented in this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

Date:

Place: Roorkee Himanshu Makkar

CERTIFICATE

This is to certify that the statement made by the candidate in declaration is correct to the best of my knowledge and belief.

Date

Place: Roorkee

(**Dr. Durga Toshniwal**)

Associate Professor

Department of Computer Science and Engineering

Indian Institue of Technology

Roorkee, India

ACKNOWLEDGEMENTS

I would first like to thank my guide Dr. Durga Toshniwal, Associate Professor, Department of Computer Science and Engineering, Indian Institue of Technology, Roorkee, for her thoughtful encouragement and careful supervision during my research work. Her unwavering enthusiasm for Data Mining kept me constantly engaged with my research and her personal generosity helped make my time at IITR enjoyable.

I am also thankful to all the staff members of the department of computer science for their support. I sincerely thank my family and all my friends for directly or indirectly helping me and giving me moral support that motivated me during the course of the work.

Abstract

Frequent itemset mining is a field of data mining wherein we extract frequent itemsets from the dataset. This may reveal sensitive patterns. Privacy Preserving Data Mining(PPDM) approaches are used to hide sensitive information from the dataset but they also reduce the utility of the dataset. Heuristics-based PPDM approaches remove the sensitive patterns from the transactions containing them, based on some heuristics. Heuristic-based approaches are simple and take lesser computational time as compared to the border-based and exact approaches. Hence they have been given much attention by researchers for exploring better heuristics that can preserve the utility of data to a great extent. In this work, we have proposed two heuristics-based approaches- Removal of Closed Sensitive Itemsets with Maximum Support (MaxRCSI) and Removal of Closed Sensitive Itemsets with Minimum Support (MinRCSI). In these proposed approaches, sensitive itemsets are reduced to closed sensitive itemsets and sanitization process is carried over reduced closed sensitive itemsets. Experiments have been performed on real datasets as well as on benchmark dataset where the proposed approaches have resulted into the sanitized data with substantially better utility as compared to the existing approaches. But these sequential approaches are not able to cope up with the massive amount of data. The other two proposed approaches- Parallelized Removal of Closed Patterns with Minimum Support (MinPRCP) and Parallelized Removal of Closed Patterns with Maximum Support (MaxPRCP) are the parallel implementation of MinRCSI and MaxRCSI on spark parallel computing framework. These parallelized approaches are scalable enough for handling large dataset. Experiments performed using benchmark datasets shows that MinPRCP and MaxPRCP scales better as compared to MinRCSI, MaxRCSI, and other sequential approaches.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION AND MOTIVATION

1.1 Introduction

Data mining allows us to extract various interesting patterns from the data by using various tools and methodologies. These extracted patterns like frequent patterns, association rules, classification model, clustering are used to facilitate decision making. Recent advancements in technologies like cloud computing and distributed processing allow us to store and process the large amount of data. The task of extraction of interesting patterns from the large dataset has become easier with parallelized distributed mining algorithms. These highly scalable data mining algorithms may result in a threat to privacy. The privacy involved the ability to control what information an individual reveal about himself and to control who can access that information. In recent years privacy has gained significant concern in the field of data mining because of sensitive patterns that reside in the data may be misused during the data mining process. Privacy-preserving data mining (PPDM) is a field of data mining which deals with the hiding of sensitive/confidential information from the data in various ways [5]:

- **Privacy-Preserving Data Publishing**
  In this technique, the data is transformed or modified before giving it to the classical data mining methods such as classification so that the results of the data mining cannot reveal any personal/confidential information. One problem with this technique is how to use transformed data with classical data mining methods.

- **Changing the results of Data Mining Applications to preserve privacy**
  In this technique, the data is applied to the classical data mining applications without any transformation but the results of these applications like classification results or association rules which may compromise the privacy are suppressed.

- **Query Auditing**
  This technique is similar to the previous case of changing the result of data mining applications. Here, either the results of queries are restricted or suppressed which compromise with the privacy.

- **Cryptographic Methods for Distributed Privacy**
  This technique uses the cryptographic tools for preserving privacy when data is distributed among multiple sites. If two or more owners at multiple sites want to perform some

common functions on their data such that no sensitive information can be revealed then a variety of cryptographic protocols may be used in order to communicate among these different sites and common functions can be calculated.

One of the approaches for protecting the confidential/personal information is to encrypt the data with a key using cryptographic techniques which completely solve the privacy concern but on the other hand, it will not work in the case of data publishing scenario and hence the third party will not be able to use data for mining. These types of techniques reduce data utility and are of no use. Different PPDM algorithms for data publishing have been devised in the recent years. Most of them transform the data such that sensitive information cannot be extracted from it.

1.2 Privacy Preserving Data Mining in Frequent Pattern Mining

Frequent Pattern Mining (FPM) is the field of Data Mining which is used to determine which things go together in transaction dataset. The prototypical example is determining what things go together in a shopping cart at the supermarket, the task at the heart of market basket analysis. Retail chains can use FPM to plan the arrangement of items on store shelves or in a catalog so that items often purchased together will be seen together[10]. Despite its benefit, FPM can also pose a threat to privacy and information security if not done or used properly. Consider the scenario where two or more companies have a very large dataset of records of their customers' buying activities. These companies decide to cooperatively conduct FIM on their datasets for their mutual benefit since this collaboration brings them an advantage over other competitors. However, some of these companies may not want to share some strategic itemsets hidden within their own data (also called restrictive itemsets) with the other parties. They would like to transform their data in such a way that these restrictive itemsets cannot be discovered[8].

1.2.1 Hiding Sensitive Pattern in Frequent Pattern Mining

Sensitive Pattern Hiding (SPH) or sanitization of the dataset is the process of removal of sensitive patterns that can be extracted from the dataset by transforming the dataset and transformed dataset is commonly called as sanitized dataset. These SPH approaches remove the occurrence of sensitive patterns from the required number of transactions to make them infrequent. Some of SPH approaches introduces the noise to the data for hiding the sensitive itemsets. This transformation process also affects the non-sensitive frequent patterns because of which some of them may also

become infrequent and hence utility of dataset decreases. Optimal sanitization of dataset with least side-effects is NP-hard problem.

1.2.2 Research Challenges

The objective of the SPH approaches is to transform the data in order to hide all the sensitive patterns from the data which are provided by data owner. The transformation process also has the side-effects on the data like generation of newly artificial patterns, removal of non-sensitive frequent itemsets etc. These side-effects reduce the utility of the data. If we are not able to extract meaningful patterns from the sanitized data then it is of no use. Thus the main challenge is to find an approach that hides all the sensitive patterns from the dataset while minimally affecting non-sensitive patterns from the data.

The transformation of dataset before publishing is considered as an overhead to the process of frequent pattern mining because of the additional time taken by SPH process. Sanitization of dataset containing large amount of transactions i.e in million requires a huge running time. So another challenge is to find an approach that require less running time on large data.

1.3 Motivation

Collaborative data mining is used when two or more organizations join hands for sharing their data with each other to mine interesting patterns from other's data which may benefit the organizations. Data shared by an organization may contain sensitive pattern and if it gets misused by another party then there can be a great loss to the organization that has shared the data. Because of sensitive content in the data, owner of the data not feel safe to publish the data. If data is not released by the organization then valuable information remains hidden inside the data. Hence some approach is required which can transform the data before publishing such that sensitive patterns would not be able to mine from the data while preserving the valuable information in transformed data.

1.4 Problem Statement

The problem statement for the work presented in this report is:"*To perform privacy preserving frequent pattern mining with improved utility and running time*".

There are many existing privacy preserving data mining models which can be used to hide sensitive patterns but it greatly affects the utility of the data..The existing approaches are sequential and require the huge running time on large data. Hence,

1.4 Organization of report

The organization of rest of report is as follows: The report is organized into 5 chapters. Chapter 2 describes the existing SPH approaches along with basic concepts required in order to understand the proposed approach. Chapter 3 describes proposed techniques: Removal of Closed Sensitive Itemsets with Minimum Support (MinRCSI), Removal of Closed Sensitive Itemsets with Maximum Support (MaxRCSI), Parallelized Removal of Closed Patterns with Minimum Support (MinPRCP) and Parallelized Removal of Closed Patterns with Maximum Support (MaxPRCP). Chapter 4 describes the performance results of proposed approaches and compares them with the existing approaches. Chapter 5 concludes the report.

2. RELATED WORK

2.1 Frequent Pattern Mining

Frequent Pattern Mining (FPM) is a field of Data mining which deals with extracting of frequent itemsets from the database. The problem of frequent pattern mining was originally proposed to find frequent set of items which are bought together in market basket data. Frequent Pattern Mining is useful in mining associations, correlations and many other interesting relationships.

Transaction: $I=\{i_1,i_2,...i_m\}$ is the set of m elements called items and $T=\{t_1, t_2,..., t_n\}$ is a set of n subsets of items called transactions. Each transaction in T is subset of I.

Support: The support of an itemset X is defined as the number of transactions which contains itemset X.

Frequent Itemsets: Frequent Itemsets are the itemsets which are having support greater than user-specified threshold.

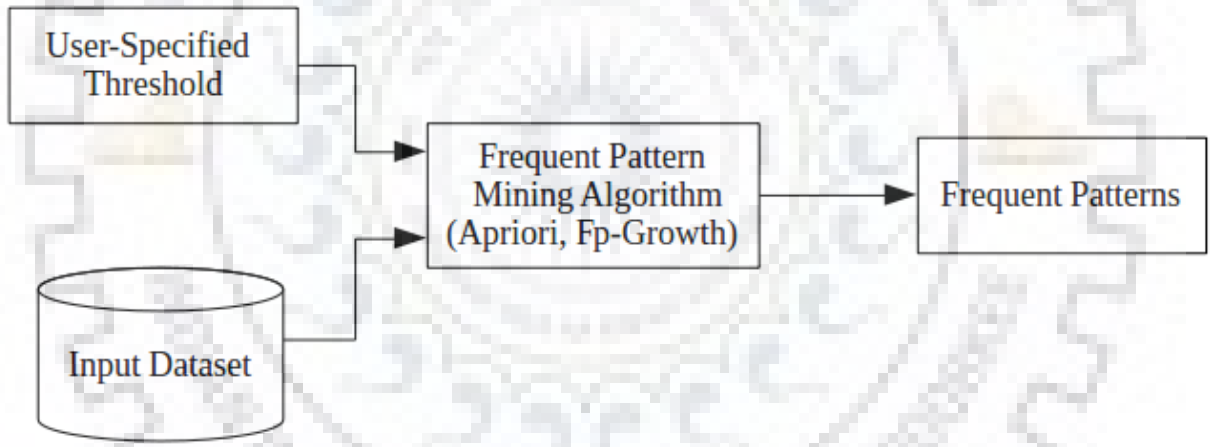

Fig. 2.1 Frequent Pattern Mining Process

Apriori algorithm is the most popular algorithm for mining frequent patterns which is designed to be applied on transaction database. Apriori takes as input i) user-specified threshold ii) dataset containing a set of transactions and outputs all frequent patterns i.e group of items supported by more than user-specified threshold number of transactions.

2.2 Sensitive Pattern Hiding

FPM can also pose a threat to privacy and information security if not done or used properly. Consider the scenario where two or more companies have a very large dataset of records of their customers' buying activities. These companies decide to cooperatively conduct FIM on their

datasets for their mutual benefit since this collaboration brings them an advantage over other competitors. However, some of these companies may not want to share some strategic itemsets hidden within their own data (also called restrictive itemsets) with the other parties. They would like to transform their data in such a way that these restrictive itemsets cannot be discovered[8]. Sensitive Pattern Hiding (SPH) is a field of data mining which provides the ways to prevent sensitive itemsets present inside the data from getting revealed. The key idea of sensitive pattern hiding algorithms is to make the support count of sensitive itemsets to less than the user-specified threshold so that they cannot appear in the result of frequent itemset mining. For this purpose, the dataset is transformed either by deleting the occurrence of sensitive itemsets from the transactions supporting them or by adding noise to the dataset.

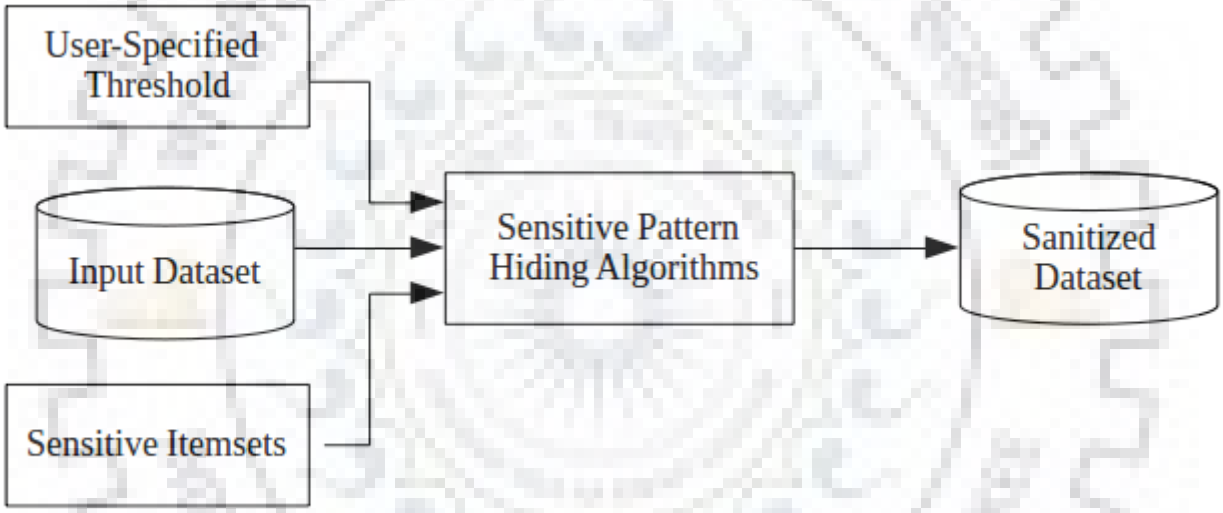

Fig. 2.2 Sensitive Pattern Hiding Process.

Sensitive Pattern Hiding Algorithms can be broadly classified into three different categories: border-based, exact approaches and heuristic-based approach. The goals of all of the sensitive pattern hiding algorithm are i) to hide maximum number of sensitive patterns ii) to reduce the side effect caused by hiding of sensitive itemsets. Side-effects of SPH algorithms involves number of non-sensitive itemsets affected by hiding process, number of falsy frequent itemsets which are generated after sanitization, etc.

Border Based approaches hide the sensitive patterns by transforming the borders in the lattice of frequent and infrequent patterns of dataset. It transforms the data such that it has minimal impact on the border to facilitate the hiding of sensitive patterns. Exact approaches identify an optimal solution that minimally affects the original dataset and causes no side-effects to hiding process by

reducing the problem of sensitive pattern hiding to constraint satisfaction problem (CSP). These approaches are slower because of huge computation to solve CSP by using linear programming.

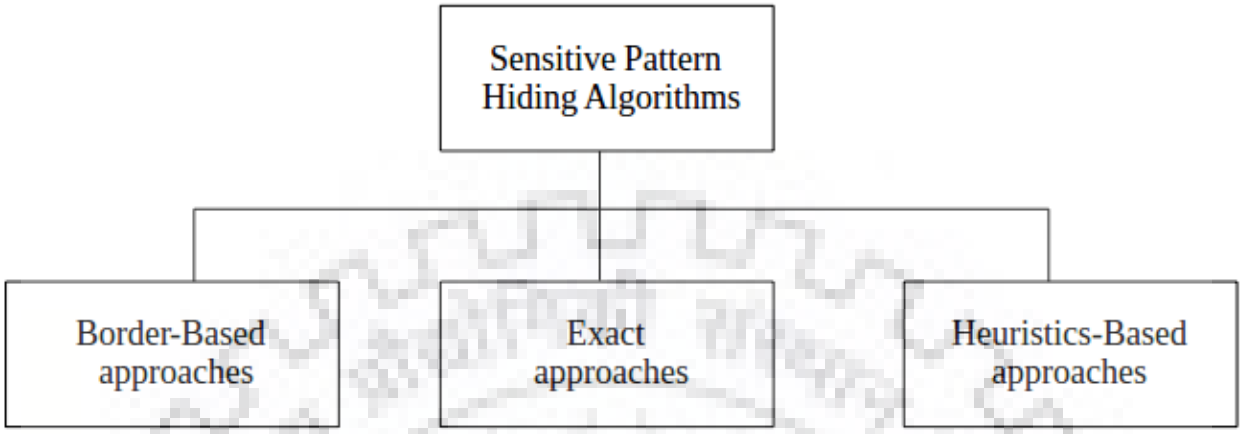

Fig 2.3 Classes of Sensitive Pattern Hiding Algorithms

Heuristic-based approaches are based on certain goals which are locally optimal and seems to be globally optimal but do not guarantee optimal hiding of sensitive patterns. These approaches can be used to transform the data in lesser time as compared to other two approaches. These are fast, efficient and scalable approaches for hiding the sensitive patterns. These approaches greatly affect the utility of the dataset. Because of its scalability and simplicity, they are one of the most popular choices for sanitization and researchers are paying more attention to these approaches for exploring some better heuristics that can preserve the utility of the data as maximum as possible.

2.3 Metrics for Performance Analysis

The objective of sensitive pattern hiding approaches is to hide the sensitive patterns from the data which also causes the side effects to the data. Fig 2.4 i) shows the frequent patterns F before sanitization which includes sensitive patterns S and non-sensitive patterns NS. Fig 2.4 ii) shows the frequent patterns F' after sanitization which does not include some non-sensitive patterns and includes some sensitive patterns and newly generated patterns. There are various measures to analyze the performance of SPH approaches which are as follows:

1. Hiding Failure: It is measured as the percentage of sensitive patterns that can be discovered after sanitization.

$$Hiding\ Failure = \frac{|F' \cap S|}{|S|} x\ 100$$

2. Misses Cost: It is measured as the percentage of non-sensitive patterns that cannot be discovered after sanitization.

$$Misses\ Cost = \frac{|F - S| - |F' - S|}{|F - S|} x\ 100$$

3. Artificial Patterns: It is measured as the percentage of newly generated patterns that can be discovered after sanitization.

$$Artificial\ Patterns = \frac{|F'| - |F' \cap F|}{|F'|} x\ 100$$

4. Utility Ratio: It is similar to Misses Cost. It is measured as the percentage of non-sensitive patterns that can be discovered after sanitization.

$$Utility\ Ratio = \frac{|F' \cap NS|}{|NS|} x\ 100$$

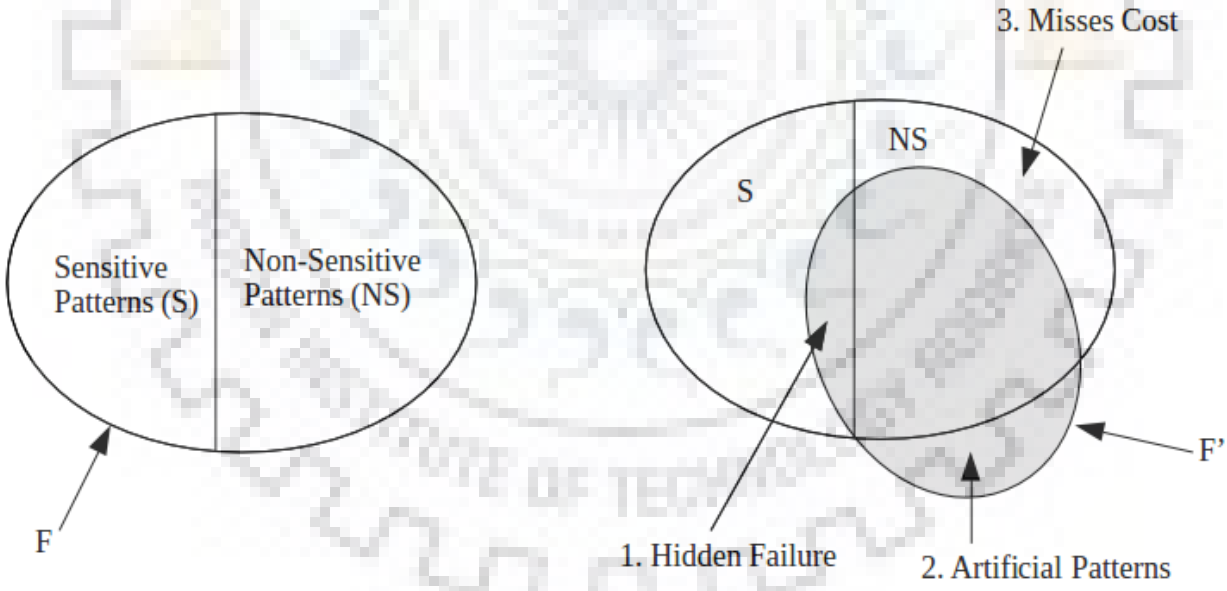

Fig 2.4 i) Frequent Patterns before Sanitization ii) Frequent Patterns after Sanitization

2.4 Literature Review

Privacy-Preserving Data Mining (PPDM) approaches for removal of sensitive itemsets from dataset can be broadly classified into three different categories namely exact, border-based and heuristics-based approaches. Heuristic-based approaches make greedy decisions at each step which are locally optimal but may not always be globally optimal. These greedy decisions are based on some heuristics and yields a good approximate of optimal solution in reasonable time. These approaches are efficient, fast and scalable solution for sanitization of dataset. Atallah et al. [12] proposed a heuristics-based technique to hide the sensitive itemsets by reducing the support. The authors proposed the construction of lattice-like graph in the database through which greedy iterative traversal is made for identifying and hiding maximum frequent item related to the sensitive rule. The author also proved that optimal hiding of sensitive itemsets is NP-hard problem by reducing the problem of hiding of sensitive itemsets to the hitting set problem. Oliveira & Zaïane [2] introduced the approach for hiding multiple itemsets from a dataset in two scans. They introduced the concept of the inverted file index for fast retrieval of the supporting transactions. Three strategies-MaxFIA, MinFIA and IGA were proposed. In Maximum Frequent Itemset Algorithm (MaxFIA), for each sensitive itemsets, supporting transactions are identified first using inverted file index and are sorted according to degree of conflict. Then it selects the required number of transactions and deletes victim item from these transactions. Victim item is the item in sensitive itemset with maximum support. MinFIA is similar to MaxFIA, difference is only in the selection of the victim item from sensitive itemset. In MinFIA victim item is the item in sensitive itemset with minimum support. IGA group the sensitive itemsets according to the itemset they share and victim item is item in itemset which is shared by all itemsets of the group and has minimum support. In [3], confidence based approach is proposed in which the confidence of an association rule is decreased which reduces the side effect caused to dataset after sanitization but it does not guarantee to hide all sensitive patterns. In [4], another heuristics-based approach is discussed which is similar to MinFIA. In this approach, number of non-sensitive pattern supported by transactions are identified and supporting transactions are identified for each sensitive rule and sorted according to number of non-sensitive pattern it contains. Then it selects the required number of transaction and removes victim item from it. Victim item in this approach is the item which belongs to the sensitive itemset having highest support. In [6], three algorithms Hidden-First, Non-Hidden-First and Hiding sensitive patterns completely with minimum side effects on non-sensitive patterns are proposed which first construct a sanitization matrix based on sensitive itemset. Sanitization matrix consists of values -1, 0 and 1. Sanitization matrix is constructed in such a way that when it is multiplied by

input dataset matrix then support of the sensitive pattern decreases. In [7], the author proposed an approach which aimed to hide all sensitive patterns and minimally affecting non-sensitive patterns. In [8], a heuristic approach SWA is discussed which hides all the sensitive itemsets in a single pass. It deals with the group of transactions of specified size (forming a window of specified size). Non-sensitive transactions are removed and copied directly to sanitized database. After that for each sensitive itemset, item with highest frequency is identified and required number of transactions are selected from supporting transactions and item identified before is removed from transactions. In [9] three approaches (Aggregate, Disaggregate and Hybrid) have been proposed. In Aggregate approach, a transaction which supports large number of frequent sensitive itemsets and less number of frequent non-sensitive itemsets is removed from the dataset and support of all frequent sensitive itemsets and frequent non-sensitive itemsets which are supported by that transaction is reduced by 1. This process is repeated until the support of all sensitive itemsets become less than the user-specified threshold. In Disaggregate approach, an item is selected from a transaction in dataset in such a way that removal of that item causes reduction of support of large number of sensitive frequent itemsets and reduction of support of less number of non-sensitive frequent itemsets. Hybrid approach is a combination of both approaches. It uses aggregate approach to identify sensitive transactions for deletion. But instead of deleting it uses disaggregate approach to identify an item to delete from sensitive transactions. In [1] and [19], three approaches were proposed that make use of blocking schemes in order to reduce either the support or confidence of sensitive rules.

Border based approaches modify the original borders in the lattice of the frequent and the infrequent patterns in the data set in order to hide the sensitive patterns. Sun & Yu [13] proposed an approach where they used the border of non-sensitive frequent itemsets to track the impact on the result database during the hiding process, and maintain the quality of the result database by selecting the modification with minimal impact at each step. The Exact approaches are non-heuristic algorithms which consider the hiding process as a constraint satisfaction problem (CSP) solved using integer programming or linear programming. In [14], author proposed an approach which formulates CSP which aimed to hide minimum number of transaction in order to remove sensitive patterns from the data. Border-Based and exact approaches maintain the good balance between privacy and utility by causing lesser side-effects, but on the other hand, they are complex and more time-consuming as compared to heuristic approaches. Heuristic-based approaches are simple, fast and may provide good approximate to the optimal solution. Hence, we have presented the work on heuristic-based

approach for exploring better heuristic in order to ensure the utility of data while hiding sensitive patterns from data.

3. PROPOSED WORK

Let input dataset D, minimum support threshold T, set of sensitive itemsets S. The goal of every SPH algorithm is to transform the D into D' such that frequent patterns that can be mined from D' at T should not contain any item from S. The transformation process also causes side effect to data. Some of the non-sensitive itemsets that could be mined from D at T become hidden from D'. As discussed in chapter 2, the heuristic-based approaches are scalable, fast and efficient but caused greater side-effects to the input data. So the effort has been done towards reducing the side-effects of heuristic-based approach.

The Proposed Approaches are heuristics-based approaches for sensitive patterns hiding. They are based on a common heuristic i.e the side-effects of hiding process on non-sensitive frequent patterns depend upon the number of sensitive patterns. Side-effects of hiding process increase with the number of sensitive patterns. In order to reduce these side-effects, the proposed approaches work on reducing the number of sensitive patterns before the hiding process. They group the sensitive patterns in such a way that removal of group representative of a group from the data will remove all the sensitive patterns represented by that group. These approaches have used closed characteristics of sensitive patterns for grouping. They reduce the sensitive patterns to the closed sensitive patterns

3.1 Closed Patterns

Closed Patterns/Itemsets provide a compact representation of large frequent itemsets. A frequent itemset is closed if it does not have any superset with same support count. Closed itemsets are lossless in the sense that they uniquely determine the set of all frequent itemsets and their exact frequency. At the same time, closed sets can themselves be orders of magnitude smaller than all frequent itemsets, especially on dense databases [15].

Consider sample dataset given in Table 3.1 where each row of table represents a transaction, T.id represents the transaction id and items represents the different items contained by that transactions. A, B, C, D represent the four different type of items. Support Count of different itemsets which appeared in sample dataset is given in Table 3.2, Table 3.3, Table 3,4 and Table 3.5. Support Count of the itemset refers to the number of transactions which contains the itemset. Closed Itemsets are represented in Table 3.6. These are the itemsets which do not have any superset with same support count. Support count of itemset {A} is 4 but there is an itemset {A, B} with same support count i.4, hence itemset {A} is not a closed itemset where Support count of itemset {B} is 5 and there is no

superset of it with same support count, hence itemset {B} is a closed itemset. These seven closed itemsets can determine all fifteen itemsets.

T. Id	Items
1	A, B
2	B, C, D
3	A, B, C, D
4	A, B, D
5	A, B, C, D

Table 3.1 Sample Dataset

Itemsets	Support Count
{A}	4
{B}	5
{C}	3
{D}	4

Table 3.2 Support Count of 1-Itemsets

Itemsets	Support Count
{A, B}	4
{A, C}	2
{A, D}	3
{B, C}	3
{B, D}	4
{C, D}	3

Table 3.3 Support Count of 2-Itemsets

Itemsets	Support Count
{A, B, C}	2
{A, B, D}	3
{A, C, D}	2
{B, C, D}	3

Table 3.4 Support Count of 3-Itemsets

Itemsets	Support Count
{A, B, C, D}	2

Table 3.5 Support Count of 4-Itemsets

Itemsets	Support Count
{B}	5
{A, B}	4
{B, C}	3
{B, D}	4
{C, D}	3
{A, B, D}	3
{A, B, C, D}	2

Table 3.6 Closed Itemsets

3.2 Proposed Framework

1. The first step is to calculate the support-count of given sensitive patterns from the input database.

2. Extract frequent sensitive patterns having support-count greater than or equal to minimum support threshold. Sensitive patterns having support lesser than minimum support threshold are not required to go through hiding process as they already are non-frequent.

3. Finding closed sensitive patterns from frequent sensitive patterns

4. Closed sensitive patterns undergo hiding process which involves reduction of support-count of sensitive patterns by removal of some itemsets from the input dataset and generates sanitized output dataset.

3.3 MinRCSI and MaxRCSI -Proposed Approaches

Two approaches- Removal of Closed Sensitive Itemsets with Min Support(MinRCSI) and Removal of Closed Sensitive Itemsets with Max Support(MaxRCSI) have been proposed. The idea of the proposed algorithms is to first reduce the sensitive itemsets into closed sensitive itemsets and then victim itemsets are identified for each closed sensitive itemsets in such a way that process of hiding of closed sensitive itemsets will hide all the sensitive itemsets. MinRCSI and MaxRCSI are sensitive patterns hiding approaches which outperform MinFIA and MaxFIA respectively. The following are the steps of the proposed approaches.

1. Reduction to Closed Sensitive Itemsets- Firstly, we reduce the sensitive itemsets to closed sensitive itemsets.

2. Identifying Victim Itemset- Sanitization or hiding process decreases the support count of closed sensitive itemsets by removing the occurrence of closed sensitive itemsets from the required number of supporting transactions which contain closed sensitive itemsets such until the support of closed sensitive itemsets become less than the minimum support threshold value and they would not appear in the result of FIM on transformed data. Removal of complete closed sensitive itemset from the transaction causes large distortion to the data. Hence in proposed approaches, hiding process removes a subset of closed sensitive itemset called as victim itemset instead of removing complete sensitive itemset from the supporting transactions which causes less distortion to the data.

3 Removal of Closed Sensitive Itemsets

After reducing the sensitive itemsets to closed sensitive itemsets and determining victim itemset for each closed sensitive itemset, the next step is to remove victim itemset for each closed sensitive itemsets from required number of supporting transactions. Supporting transactions of an itemset are the transactions which contained that itemset. Removal of victim itemset of a closed sensitive itemset from a supporting transaction decreases the support-count of that closed sensitive itemset by one. For hiding a closed sensitive itemset, it is required to make the support-count of it below to user-specified minimum support threshold. Initially, if the support count of closed sensitive itemset is S and minimum support threshold is *minsupp* then in order to remove closed sensitive itemset, it is required to delete victim itemset from S-T+1 number of supporting transactions. The selection of supporting transactions for removal of victim itemset is based on the concept of degree of conflict used in MaxFIA and MinFIA approach [2]. Transactions are sorted by according to number of closed sensitive itemsets supported by the transactions also called degree of conflict. Transaction supporting the maximum number of closed sensitive itemsets is chosen first for removal.

3.4 MinPRCP and MaxPRCP- Proposed Parallelized Approaches

Two approaches- MinPRCP- Parallelized Removal of Closed Patterns with Minimum Support and MaxPRCP- Parallelized Removal of Closed Patterns with Maximum Support have been proposed. MinPRCP and MaxPRCP are the parallelized versions of MinRCSI and MaxRCSI respectively. Both approaches are implemented over spark distributive framework. These approaches are designed to run in the distributed enviornment (Hadoop File System or Standalone Spark) over multiple nodes for parallel processing. Parallel execution over multiples nodes reduces the time taken by sanitization process.

4. EXPERIMENTS AND DISCUSSION

Different experiments were conducted in order to measure the performance of proposed approaches. Proposed approaches- MinRCSI and MaxRCSI aim to reduce the side-effects of sanitization on the data whereas proposed approaches- MinPRCP and MaxPRCP are the parallelized implementation of MinRCSI and MaxRCSI on Spark Framework which aim to reduce the time taken by sanitization approach on the large data. Two different types of experiments were conducted for measuring the performance of two different types of approaches. The first type of experiments were done to analyze the running time of the approaches on large data. The second type of experiments were done to analyze the efficiency of the different approaches. It includes the comparison of Utility Ratio, Hiding Failure and number of artificial patterns of proposed approach with existing approaches. MinPRCP and MaxPRCP will have same efficiency as of MinRCSI and MaxRCSI respectively. So there is no need to compare the efficiency of MinPRCP and MaxPRCP. All the experiments were conducted on the single Ubuntu workstation having 48 cores, 64GB memory, running Hadoop version 2.7 with spark version 2.2.0 in a standalone mode. We have compared the performance of proposed approaches with earlier approaches [2] - MinFIA and MaxFIA. In [2], experimental results showed that MinFIA and MaxFIA are equally efficient. MinRCSI aims to improve the efficiency of MinFIA and MaxRCSI aims to improve the efficiency of MaxFIA. We have compared the performance of proposed approaches on different types of dataset which are described in section 4.1.

4.1 Description of Dataset

Two different types of datasets used are- Benchmark Datasets (BD) and Real-world Datasets.

4.1.1 Benchmark Synthetic Dataset

Benchmark Dataset Generator (BDG) is used for generating the dataset containing market basket data. BDG generates the dataset depending upon the different parameters supplied to it. These different parameters include – number of transactions, number of different items and average length of each transaction. Five different datasets were generated with the different parameters described in Table 4.1. As we have used the spark framework for reducing the time taken by sanitization approaches on the large datasets so we have taken the number of transactions parameter in millions while generating the BDs. Only in one (1^{st}) synthetic dataset, we have taken less number of

transactions. This synthetic dataset is for analyzing the efficiency of MinRCSI and MaxRCSI. Pre-processing on the data generated by BDG were also done before applying the SPH approaches.

4.1.2 Real-World Dataset

Three different real-world datasets were used in order to compare the efficiency of proposed approaches. These datasets include– Chess Dataset, Accident Dataset and Connect Dataset. The description of these datasets is given in Table 4.2..

BD No.	Number of Transactions	Number of Items	Average Transaction Length
1	100,000	50	15
2	5,000,000	500	40
3	7,500,000	500	40
4	10,000,000	500	50
5	20,000,000	500	50
6	25,000,000	500	50

Table 4.1 Description of Benchmark Datasets

No.	Name	Number of Transactions	Different Items	Average Tramsaction Length
1	Chess Dataset	3,196	75	37
2	Accident Dataset	340,183	572	45
3	Connect Dataset	67,557	127	43

Table 4.2 Description of Real-World Datasets

4.2 Analyzing the Utility of Proposed Approach-MinRCSI

In this section, we will discuss the results obtained by MinRCSI on different datasets and will compare it with MinFIA. The MinRCSI approach deals with hiding of each and every sensitive itemsets and has 0% hidden failure. In this approach, sanitization of data is done by removal of sensitive itemsets from the data and no new data is introduced during sanitization. So this approach does not produce any artificial patterns. But removing of sensitive itemsets from data may remove non-sensitive itemsets. So we have used utility ratio as a measure of performance which can be

measured as the percentage of non-sensitive frequent itemsets that can be mined after sanitization as described in section 2.2.1.

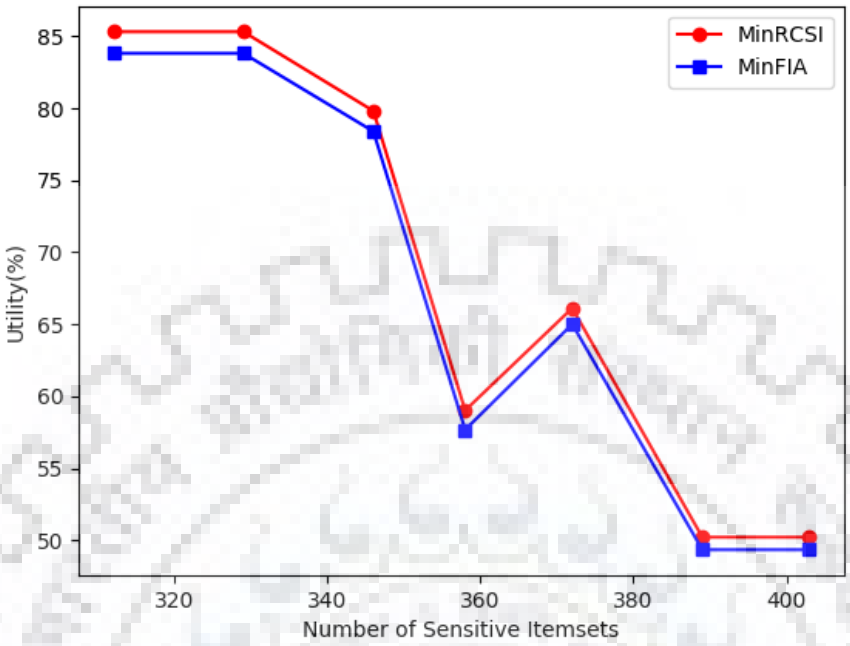

Fig. 4.1 Performance of Proposed MinRCSI on Chess Dataset

Figure 4.1 describes the utility results obtained by MinRCSI on chess dataset and comparison of them with the utility results obtained by MinFIA. It can be seen that there is an improvement of 1-2% in utility ratio. There are 6,439,702 number of frequent itemsets which can be extracted at 40% minimum support threshold from chess dataset. MinRCSI saves 64,397 (approx.) more number of non-sensitive frequent itemsets than MinFIA from getting hidden as a side-effect of the sanitization.

Fig 4.2 describes the Utility Results obtained by MinRCSI and MinFIA on accident dataset. Here the set of sensitive itemsets are chosen randomly and size of the set varying from 100 to 200. In accident dataset also, MinRCSI has shown 1-2% higher utility ratio than that of MinFIA. We have chosen minimum support threshold equal to 40% and there are 32,528 number of frequent patterns which can be extracted from the accident dataset at this minimum support threshold value. MinRCSI saves 487 (approx.) more number of non-sensitive frequent itemsets than MinFIA from getting hidden as a side-effect of the sanitization. In this set of experiments also, utility ratio has been decreased with the increase of the number of sensitive itemsets chosen for sanitization process. Utility ratio decreases with the fast rate when number of sensitive itemsets chosen is less than 148 and for greater than 148, it decreases with the slower rate. MinPRCP will also have the same utility results as of MinRCSI but their running time will vary which is discussed in section 4.3.

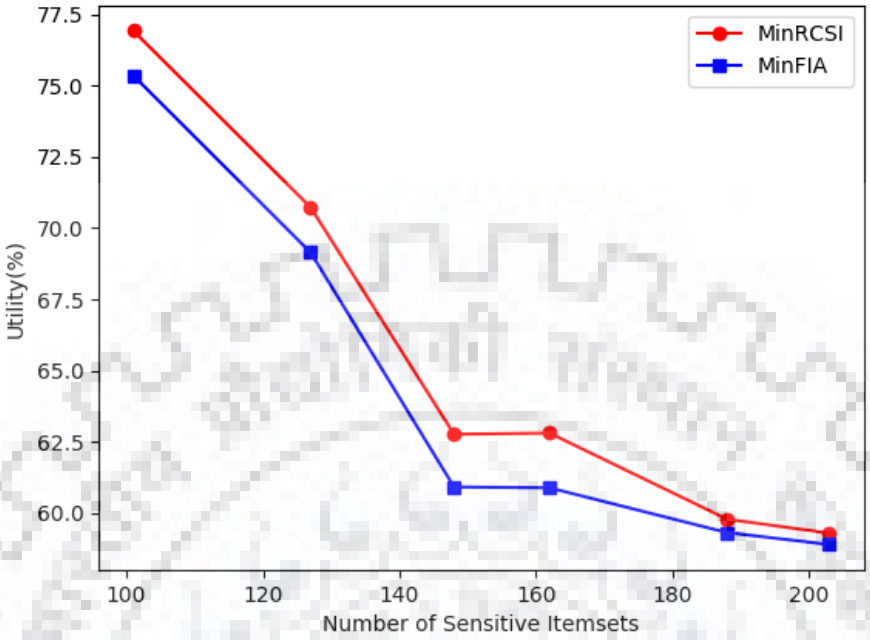

Fig 4.2 Performance of Proposed MinRCSI on Accident Dataset

Figure 4.3 describes the performance of MinRCSI on connect dataset. We have chosen minimum support threshold equal to 85% and there are 142,127 number of frequent patterns which can be extracted from the accident dataset at this minimum support threshold value. Utility Results obtained from this dataset are similar to the utility results obtained from accident dataset. In this dataset also, MinRCSI has shown better utility results as compared to MinFIA. Utility decreases with the fast rate when the number of sensitive itemsets was less than 1676 and for greater than 1676, it decreases ar the slower rate.

Figure 4.4 describes the performance of MinRCSI on Synthetic dataset. We have chosen 1st synthetic dataset consists of 100,000 number of transactions described in table 4.1. Minimum Support Threshold is set to 10%. The total number of frequent patterns that can be extracted at this minimum support threshold is 1996. Utility ratio is measured with different number of sensitive itemsets varying from 40-160. Utility Ratio is decreasing with the increase of number of sensitive itemsets but initially with slow rate with lesser number of sensitive itemsets (less than 119) and for greater than 119, it decreases with a faster rate. Utility Ratio depends upon the type of data and the type of set of sensitive itemsets. If there are large number of closed itemsets in sensitive itemsets then less side-effect will be caused to non-sensitive itemsets. Hence, the difference in rate of change

of utility with number of sensitive itemsets can be seen in various experiment but generally, it has been observed that utility ratio decreases with the increase of number of sensitive itemsets. After analyzing the performance of both approaches on different datasets, it can be seen that proposed MinRCSI performs better than MinFIA.

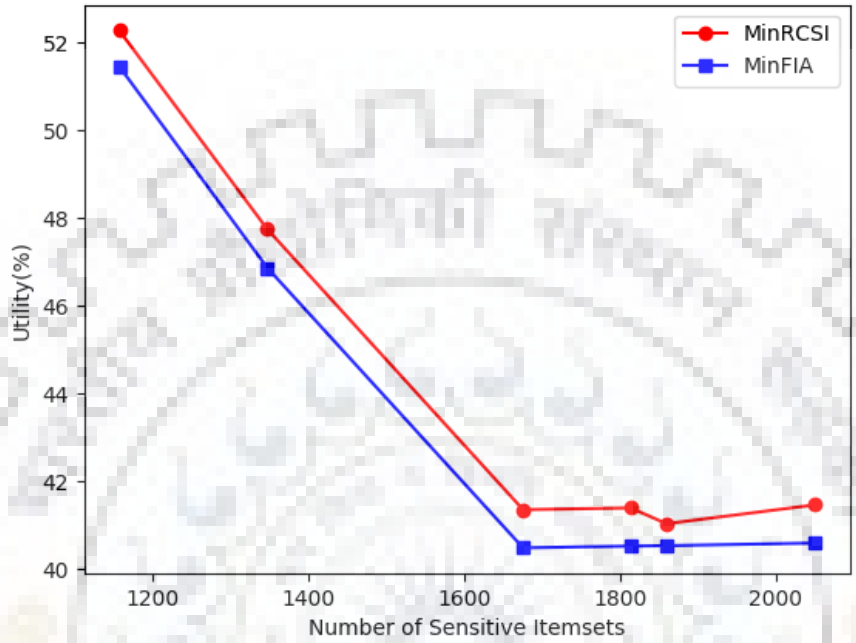

Fig 4.3 Performance of Proposed MinRCSI on Connect Dataset

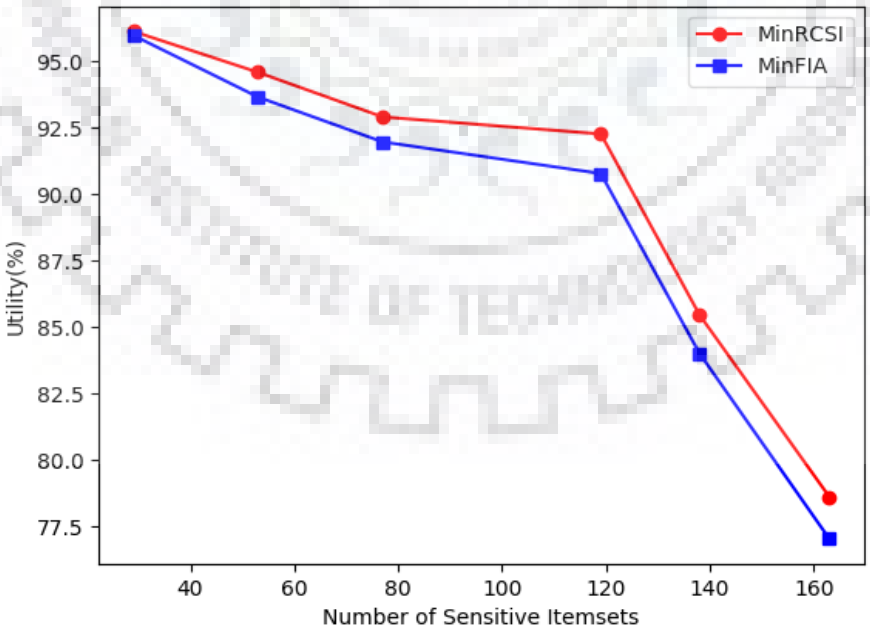

Fig 4.4 Performance of Proposed MinRCSI on Synthetic Dataset

4.3 Analyzing the Utility of Proposed Approach-MaxRCSI

In this section, we will discuss the results obtained by proposed MaxRCSI on different datasets and will compare it with MaxFIA. The proposed MaxRCSI approach also has 0% hidden failure and 0% artificial patterns. Utility results of MaxRCSI is compared with utility results of MaxFIA on accident, chess, connect and synthetic dataset. Fig 4.5 describes the comparison of utility results obtained from MaxRCSI with the utility results obtained from MaxFIA on chess Dataset. The minimum support threshold was set to 40%. Number of sensitive itemsets vary from 320 to 400. Utility results obtained from proposed MaxRCSI approach are 1-5% greater than the utility results obtained from MaxFIA. This means proposed MaxRCSI saves 64397-321985 (approx.) more number of non-sensitive frequent itemsets as compared to MaxFIA from getting hidden as the side-effect of the sanitization. In this set of experiments also, utility ratio decreases with the increase in number of sensitive itemsets.

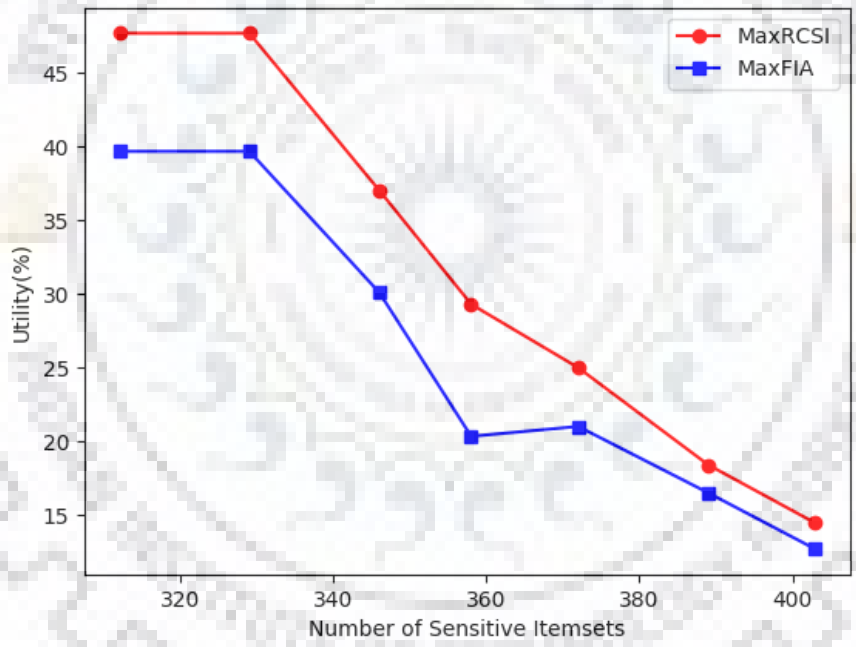

Fig 4.5 Performance of Proposed MaxRCSI on Chess Dataset

Fig 4.6 describes the comparison of utility results obtained from MaxRCSI with the utility results obtained from MaxFIA on accident dataset. The minimum support threshold was set to 40%. Number of sensitive itemsets vary from 100 to 200. Utility results obtained from proposed MaxRCSI approach are 2-8% greater than the utility results obtained from MaxFIA. This means proposed MaxRCSI saves 650-2602 (approx.) more number of non-sensitive frequent itemsets as

compared to MaxFIA from getting hidden as the side-effect of the sanitization. In connect dataset also, proposed MaxRCSI has shown better utility results than MaxFIA as shown in Fig 4.7.

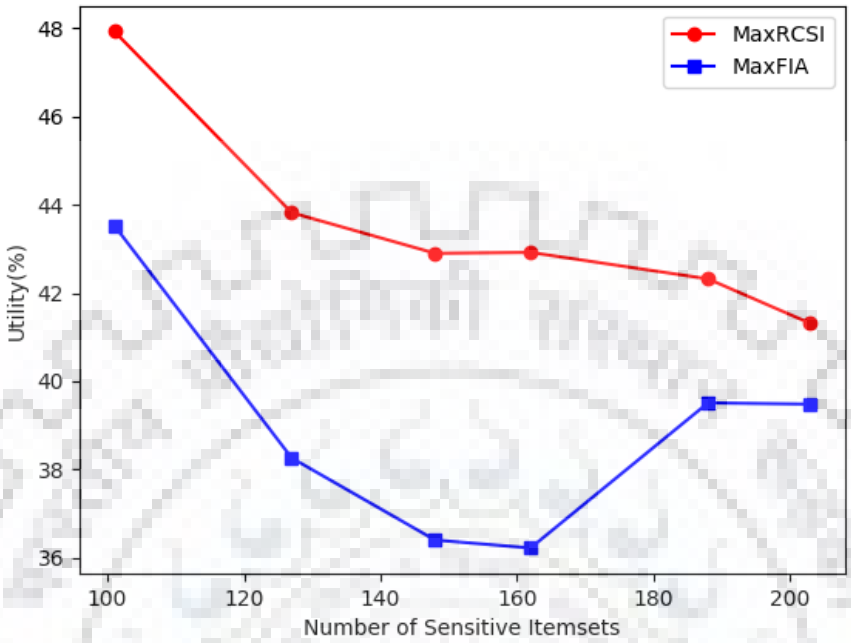

Fig 4.6 Performance of Proposed MaxRCSI on Accident Dataset

The minimum support threshold was 85%. Proposed MaxRCSI saves 2131-1.5% (aprox) more number of non-sensitive frequent itemsets from getting hidden.

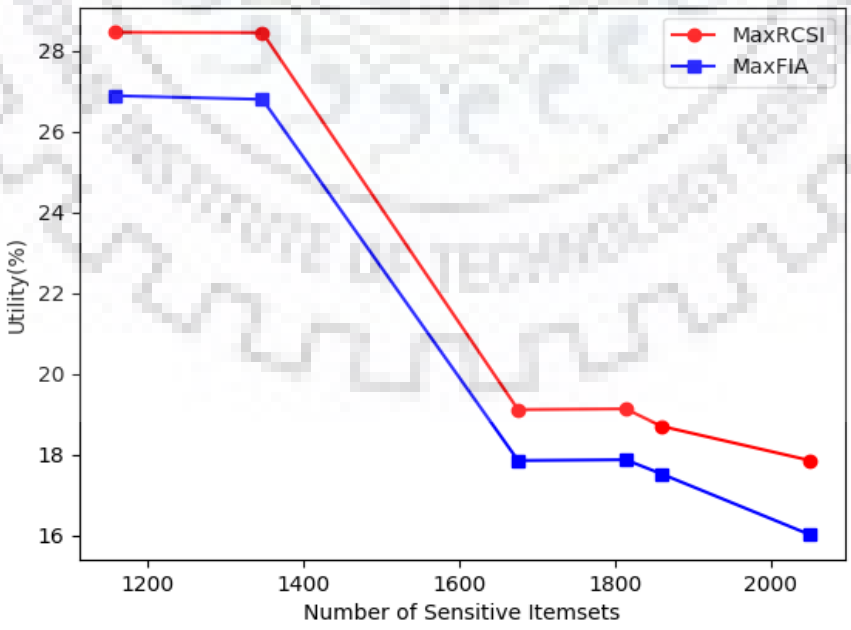

Fig 4.7 Performance of Proposed MaxRCSI on Connect Dataset

In synthetic dataset, proposed MaxRCSI has similar utility results as of MaxFIA shown in Fig 4.8. Benchmark dataset (1st) having 100,000 number of transactions with 10% of minimum support threshold was used for conducting this set of experiments. Proposed MaxRCSI saves 9-0.5% (aprox) more number of non-sensitive frequent itemsets from getting hidden. Utility Ratio as discussed before, depends upon the type of dataset and type of sensitive itemsets chosen. This benchmark dataset might have lesser number of closed itemsets as compared to other datasets. This can be the reason for similar utility results of proposed MaxRCSI and MaxFIA. Other reason for similar results can be the less number of closed itemsets in chosen set of sensitive itemsets. After analyzing the performance of both approaches on different datasets, it can be seen that proposed MaxRCSI performs better than MaxFIA.

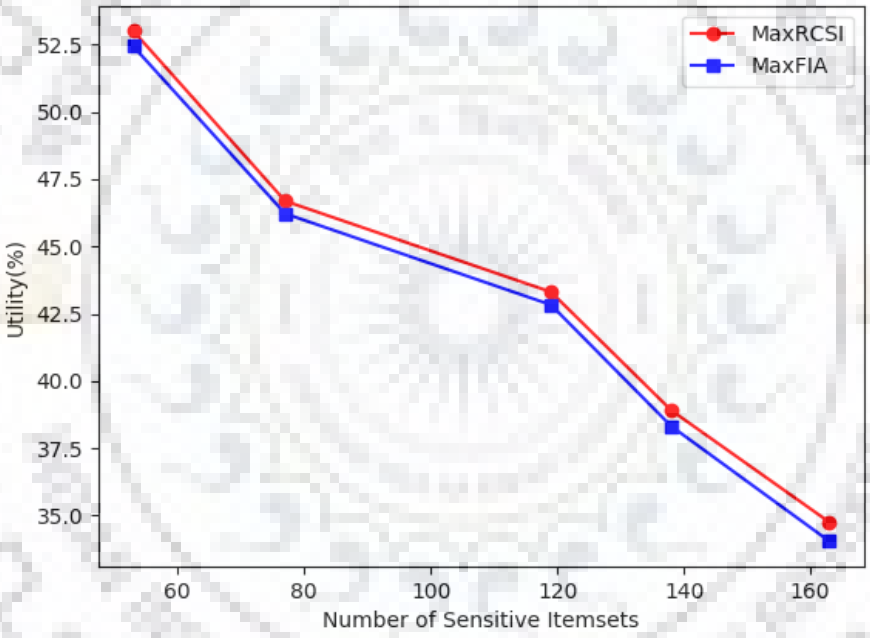

Fig 4.8 Performance of Proposed MaxRCSI on Synthetic Dataset

4.4 Analyzing the effect of Minimum Support Threshold on Proposed Approaches—MinRCSI and MaxRCSI

This set of experiments were done in order to analyze the effect of minimum support threshold (MST) on the performance of MinRCSI and MaxRCSI approach. The performance of both proposed approaches is studied at different chosen value of MST. Number of transactions that proposed approaches choose while sanitization for transformation depends upon the support-count of sensitive itemset and MST value i.e *s-minsup+1* where *s* is the support-count of sensitive itemset and minsup is minimum support threshold. If we choose MST to be high, then many of the sensitive

itemsets become in-frequent at this MST value and the value *s-minsup+1* will be low. As studied in section 4.2 and 4.3, utility results of MinRCSI and MaxRCSI decreases with the increase of number of sensitive itemsets. With high value of MST, we are left with less number of frequent sensitive itemsets and also number of transactions to choose for transformation decreases, hence utility results of SPH approaches increases. Similarly if we choose MST to be low, then large number of sensitive itemsets will appear in the set of frequent itemsets and the value *s-minsup+1* will be high. The large number of transactions need to be transformed for hiding of large number of sensitive itemsets, hence utility results of SPH approaches decreases. Benchmark dataset (1st) having 100,000 number of transactions was used for analyzing the effect of MST on proposed approaches.

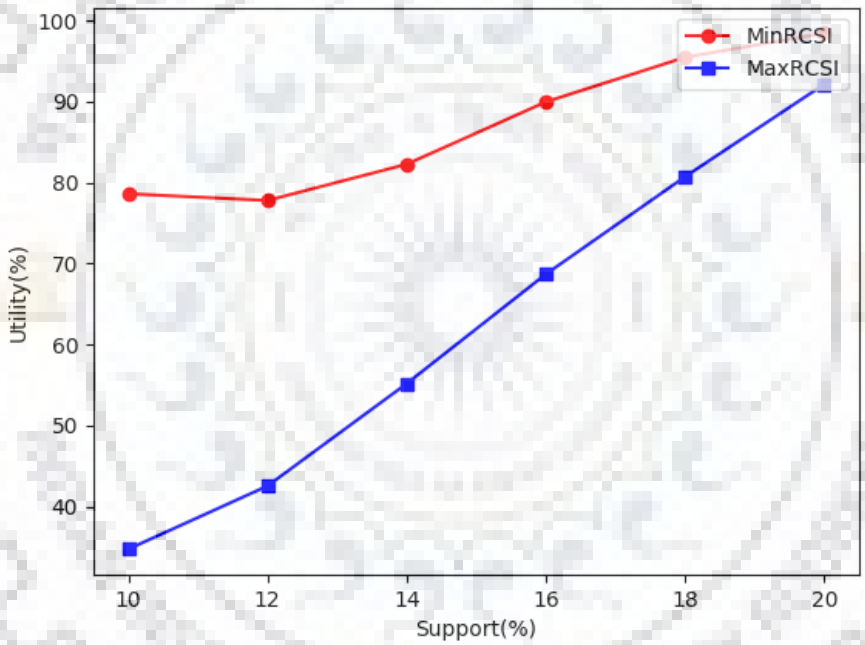

Fig 4.9 Effect of MST on Proposed Approaches -MinRCSI and MaxRCSI

Number of sensitive itemsets selected for the experiments were 163. Here we have used synthetic datasets and cannot apply any application based constraints so we have chosen the set of sensitive itemsets randomly. Fig 4.9 describes the utility results of MinRCSI and MaxRCSI approach obtained with different MST values. It can be seen that utility results of both approaches increase with the increase of MST value. Hence, SPH approaches are more efficient at higher value of MST. MinRCSI has shown relatively better utility results than MaxRCSI but the utility results of both approaches become closer at high value of MST. Both are the heuristic-based approaches and for

this benchmark dataset MinRCSI has performed better. MinPRCP and MaxPRCP will also have the same utility results as of MinRCSI and MaxRCSI respectively.

4.5 Analyzing the Running Time of Proposed Approach- MinPRCP

Proposed MinPRCP is the parallel implementation of proposed MinRCSI on Spark Framework. It is equally efficient as MinRCSI in terms of these performance metrics- hidden failure (i.e. 0%), artificial patterns (i.e. 0%) and utility ratio. The parallel implementation on spark reduces the time taken by SPH approach. Traditional SPH approaches were sequential and take time when operated on large dataset as processing the large dataset eg. sorting of data according to degree of conflict will require huge amount of running time on a single node. Hence partitioning of data across multiple and processing it in a parallel way on multiple nodes across the spark cluster saves large amount running time. This set of experiments were done in order to analyze the running time of proposed approach- MinPRCP. We have proposed MinPRCP approach as the improvement over existing MinFIA approach. That is why we have compared the running time of proposed MinPRCP with proposed MinRCSI (sequential version of MinPRCP) and MinFIA. We have studied the effect of running time of SPH approaches with respect to varying number of sensitive itemsets. With the increase of number of sensitive itemsets transformation overhead require also increases, hence running time of SPH approaches also increases. This set of experiments were conducted on a single node have 48 number of cores, 64 GB memory (executor memory- 38 GB, driver memory- 15 GB) and running standalone spark. Since we have used only single node, running time of proposed approach- MinPRCP can be reduced further if more number of nodes are used for parallel processing. Experiments were carried over different large benchmark datasets.

Fig 4.10 shows the running time of proposed MinPRCP approach on Benchmark dataset 2 (5,000,000 number of transactions) and compares it with the running time of sequential SPH approaches. Minimum support threshold was set to 25% for conducting this set of experiments. We have studied the effect of number of sensitive itemsets on SPH approaches by varying the number of sensitive itemsets from 4 to 20. As spark performs better on large dataset and when we have multiple nodes in a cluster. Spark results on single node are still relatively better than sequential approaches. MinPRCP took 150-200 sec(approx.) lesser than the other sequential approach- MinFIA and proposed MinRCSI. It can also be concluded from the results that running time of SPH approaches increases with the increase of number of sensitive itemsets.

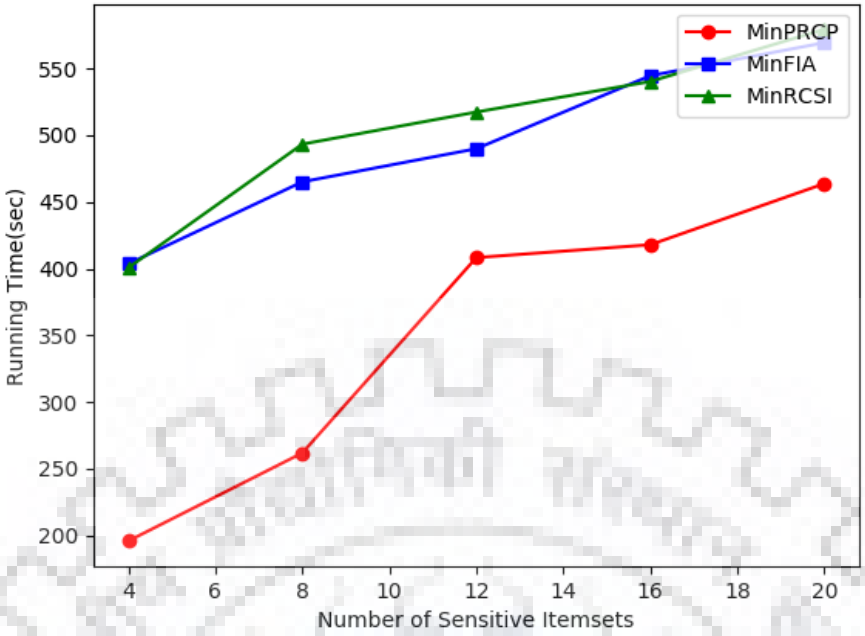

Fig 4.10 Analysis of Running Time of Proposed MinPRCP on Benchmark Dataset 2

(5,000,000 number of transactions)

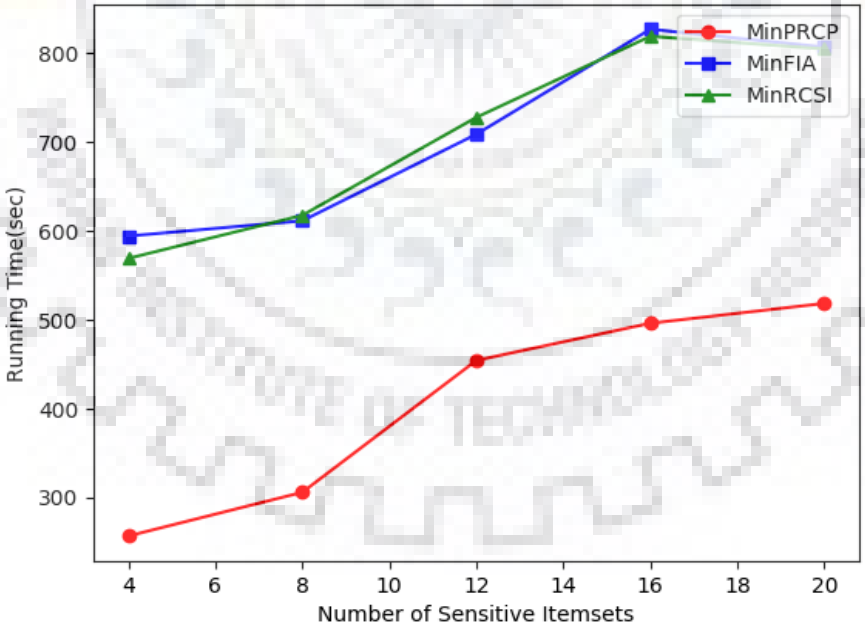

Fig 4.11 Analysis of Running Time of Proposed MinPRCP on Benchmark Dataset 3

(7,500,000 number of transactions)

Fig 4.11 shows the running time of MinPRCP approach on benchmark dataset 3 (7,500,000 number of transactions) and compares it with the running time of sequential SPH approaches- MinFIA and proposed MinRCSI. The value of MST used was 25% for this dataset. For this dataset, proposed MinPRCP took 300-350(approx.) seconds less than the sequential SPH approaches- MinFIA and proposed MinRCSI. As the size of benchmark dataset 3 is greater than benchmark dataset 2, proposed MinPRCP (spark implementation) has performed relatively better on benchmark dataset 3 as compared to benchmark dataset 2.

Fig 4.12 compares the running time of approaches on benchmark dataset 4 (10,000,000 number of transactions). MST value was similar to previous set of experiments i.e. 25%. For this dataset, proposed MinPRCP took 400-600 (approx.) seconds less than sequential SPH approaches. It can be concluded that with the increase of dataset size, time difference between parallelized and sequential approaches has increased and parallelized approach i.e. MinPRCP is performing relatively better on big dataset.

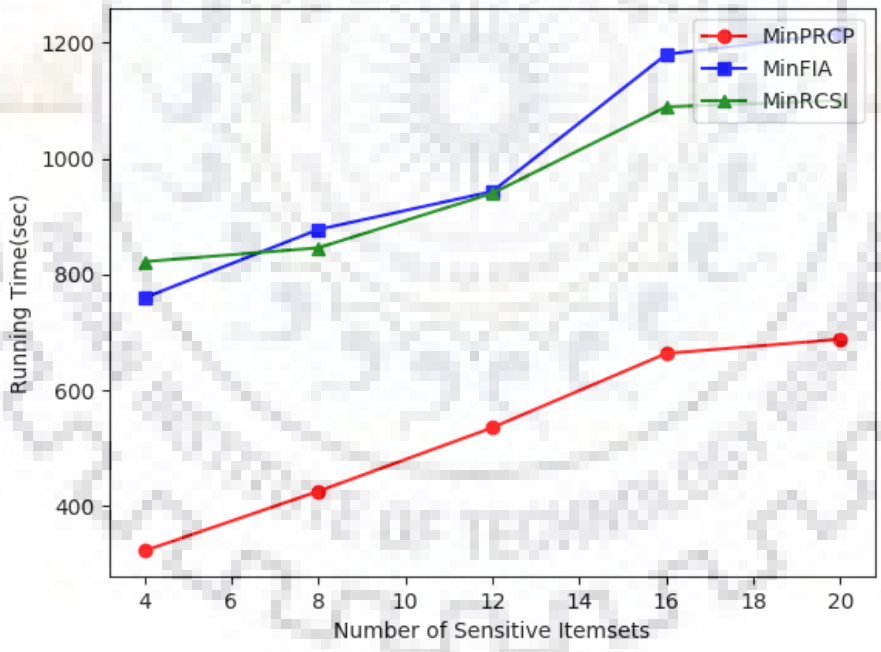

Fig 4.12 Analysis of Running Time of Proposed MinPRCP on Benchmark Dataset 4 (10,000,000 number of transactions)

To analyze the performance of proposed MinPRCP in better way, dataset size was further increased and same set of experiments were conducted on benchmark dataset 4 (20,000,000 number of transactions). It can be concluded from the Fig 4.13, sequential approaches did not scale well when number of sensitive itemsets were increased beyond 12. When the number of sensitive itemsets were less than 12, MinPRCP took 1500-2000 (approx.) seconds less than sequential SPH approaches. But when number of sensitive itemsets were increased beyond 12, running time difference between sequential and parallelized approaches increased exponentially. It can be seen that proposed MinPRCP scaled very well with large number of sensitive itemsets by taking 4000-7000 lesser running time as compared to MinFIA and proposed MinRCSI.

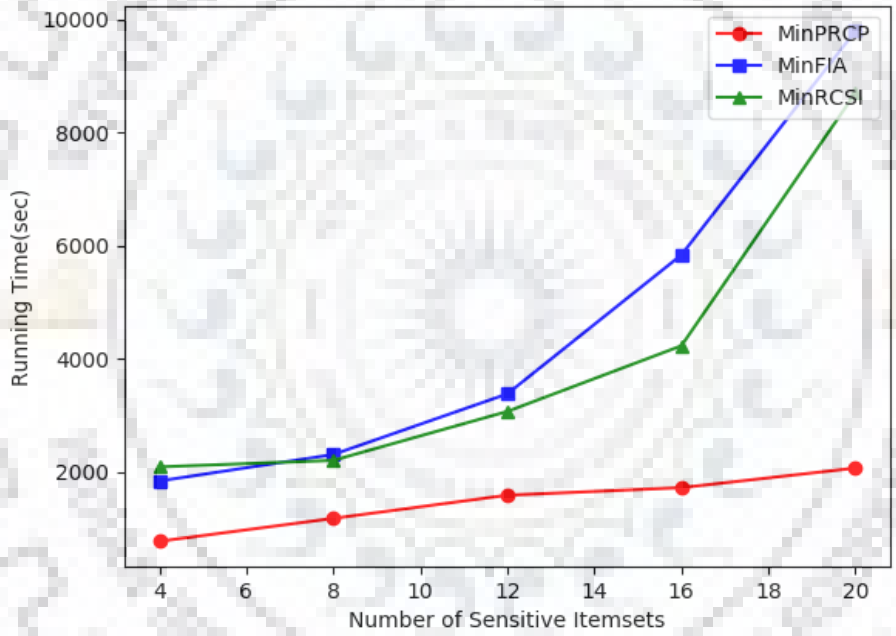

Fig 4.13 Analysis of Running Time of Proposed MinPRCP on Benchmark Dataset 5

(20,000,000 number of transactions)

The last set of experiments for analyzing the scalability of MinPRCP were conducted on Benchmark Dataset 6 (25,000,000 number of transactions). Same MST value used in earlier set of experiments i.e. 25% was used in this set of experiments also. Running time of MinFIA and proposed MinRCSI is relatively very high as compared to running time of proposed MinPRCP. Running time of MinFIA and proposed MinRCSI with chosen 20 number of sensitive itemsets for sanitization process is greater than 8000 seconds whereas running time of MinPRCP is around 3000

seconds as shown in fig. 4.14.. In all of the experiments, proposed MinPRCP has shown better scalability results even when only single spark node was used. Proposed MinPRCP will be able to scale much well if provided with more number of nodes in spark cluster.

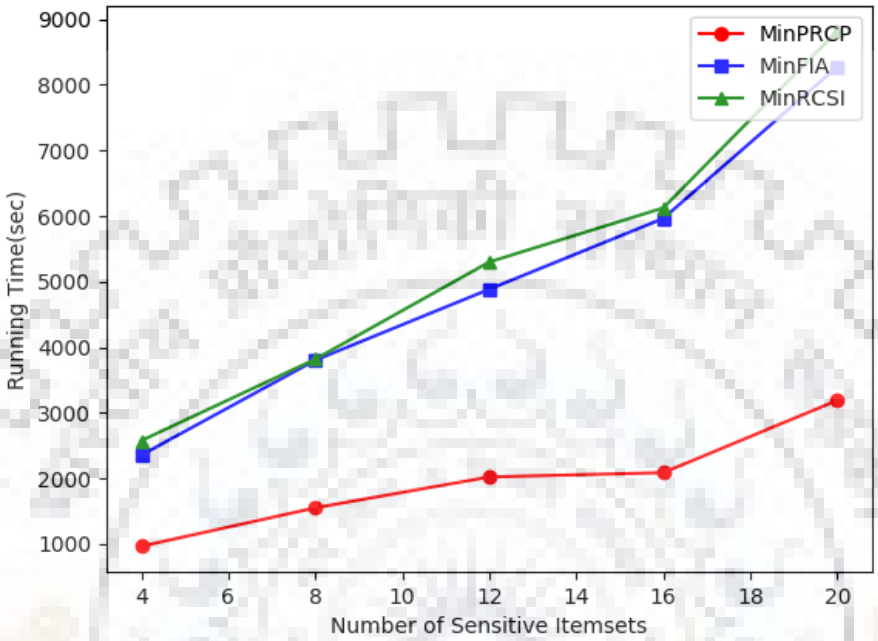

Fig. 4.14 Analysis of Running Time of Proposed MinPRCP on Benchmark Dataset 6 (25,000,000 number of transactions)

4.6 Analyzing the Running Time of Proposed Approach- MaxPRCP

Proposed MaxPRCP approach is similar to Proposed MinPRCP approach except for choosing the victim itemset. Hence, it should take similar time as Proposed MinPRCP. Same set of sensitive itemsets and MST value for corresponding datasets is used which were used in the experiments done in section 4.4. First set of experiments were done on Benchmark Dataset 2 (5,000,000 number of transactions) in order to analyze the running time of proposed MaxPRCP and results obtained were similar to the results of proposed MinPRCP. MaxPRCP took relatively lesser time as compared to MaxFIA and MaxPRCP as shown in fig.4.15. In second set of experiments also, where analysis is done on benchmark dataset 3 (7,500,000 number of transactions) proposed MaxPRCP scaled better and took lesser running time as compared to sequential approaches as shown in fig. 4.16. In third set of experiments also, where analysis is done on Benchmark dataset 4 (10,000,000 number of transactions) proposed MaxPRCP took 500-600 (approx.) seconds less than the sequential SPH approach as shown in fig 4.17.

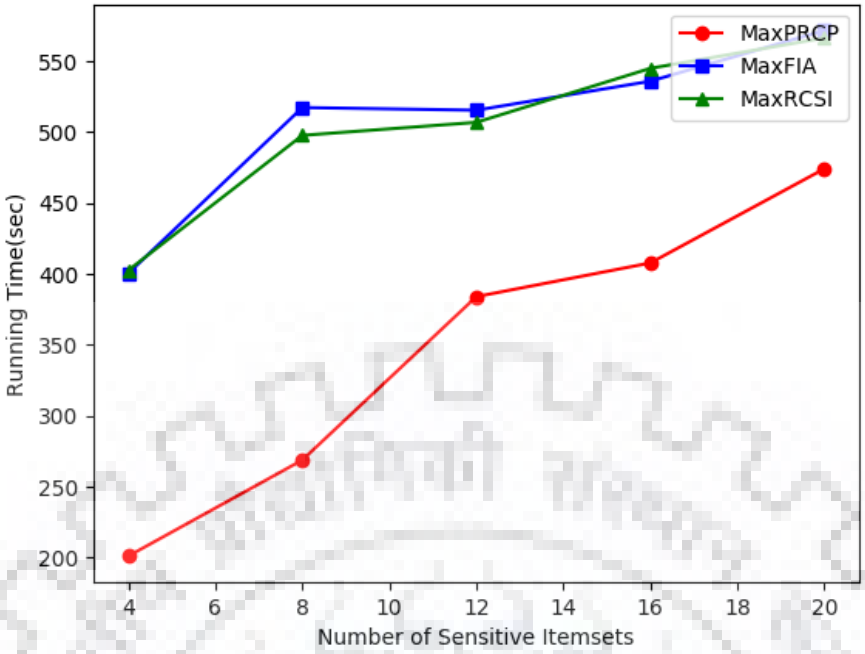

Fig 4.15 Analysis of Running Time of Proposed MaxPRCP on Benchmark Dataset 2 (5,000,000 number of transactions)

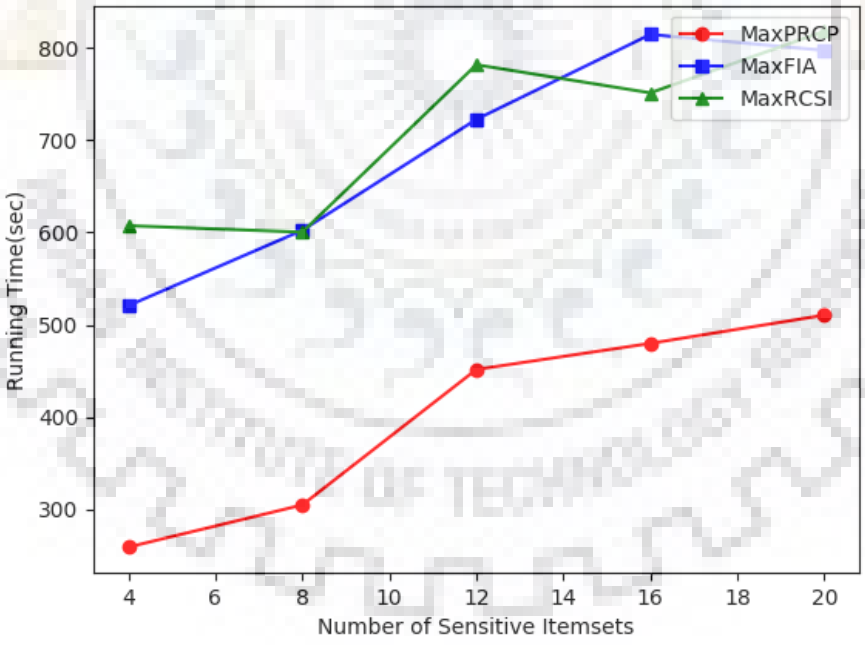

Fig 4.16 Analysis of Running Time of Proposed MaxPRCP on Benchmark Dataset 3 (7,500,000 number of transactions)

Same set of experiments were conducted on benchmark dataset 4 (20,000,000 number of transactions). When the number of sensitive itemsets was less than 12, MinPRCP took 1500-2000

(approx.) seconds less than sequential SPH approaches. But when number of sensitive itemsets were increased beyond 12, running time difference between parallelized and sequential approaches increased exponentially.

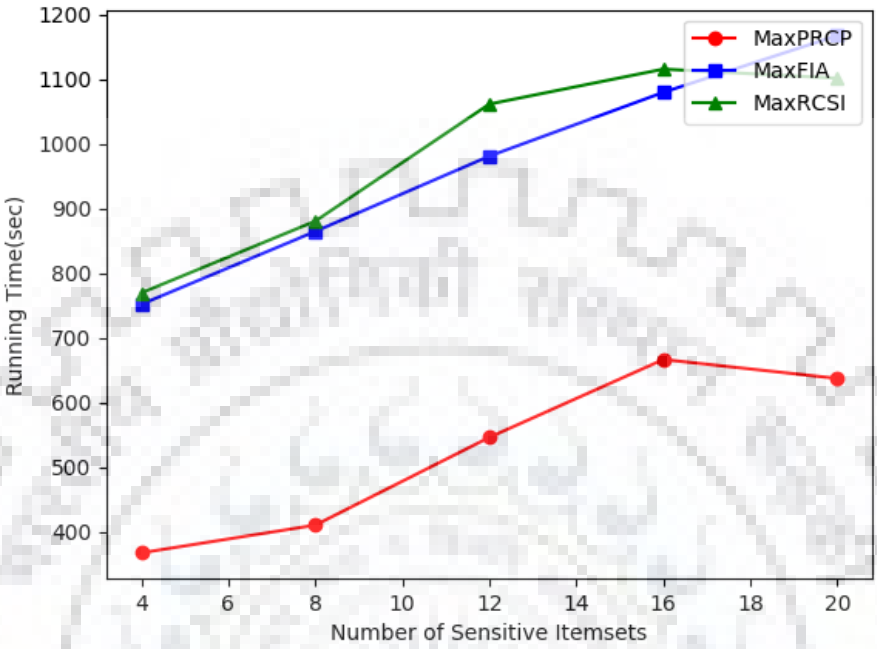

Fig 4.17 Analysis of Running Time of Proposed MaxPRCP on Benchmark Dataset 4 (10,000,000 number of transactions)

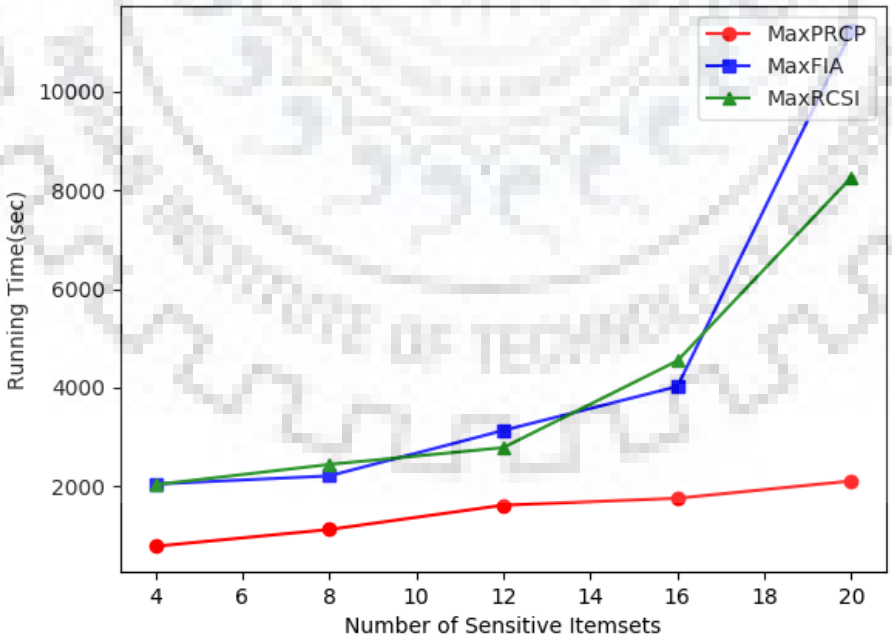

Fig 4.18 Analysis of Running Time of Proposed MaxPRCP on Benchmark Dataset 5 (20,000,000 number of transactions)

4.6 Analyzing the Running Time of Proposed Approach- MaxPRCP

In the last set of experiments, Running time of proposed MaxPRCP is compared with the running time of MinFIA and proposed MinRCSI on Benchmark Dataset 6 (25,000,000 number of transactions) as shown in Fig 4.19. In this set of experiments also, similar results were obtained. Proposed MaxPRCP was able to scale well on large dataset whereas MinFIA and proposed MinRCSI did not scale well and took relatively very large time as compared to MaxPRCP.

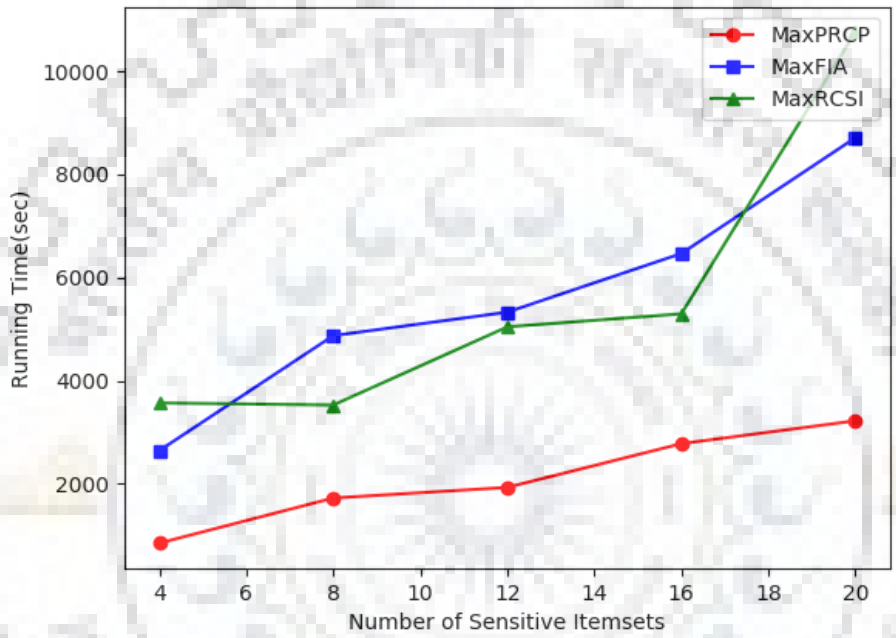

Fig 4.19 Analysis of Running Time of Proposed MaxPRCP on Benchmark Dataset 6 (25,000,000 number of transactions)

5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

Till now works that primarily targeted hiding sensitive patterns from data tried to balance two key factors, namely privacy and utility of data. Among the three broad streams of SPH approaches, namely border-based, exact and heuristic-based approaches, we focused on the latter one. The earlier duo cause significant computational cost and are not feasible for application into large dataset. Heuristic-based approaches gain advantage in the field of memory-efficiency and scalability, although they also cause large side-effects. In this work, proposed algorithms- MinRCSI and MaxRCSI, focus on reducing the side-effects by reducing large number of sensitive itemsets to closed sensitive itemsets. The proposed approaches are tested extensively by performing different experiments under varying parameters. The experiments were performed on three real datasets and one benchmark dataset. Both proposed approaches have performed relatively better with respect to their corresponding traditional approaches. Two parallelized approaches- MinPRCP and MaxPRCP have been proposed further which extend the work of MinRCSI and MaxRCSI on spark framework. These approaches scale well with large data if proper resources are available for the implementation of spark cluster. Experiments were performed on five different large benchmark datasets where the parallelized proposed approaches scaled greatly with increased load.

5.2 Future Work

The proposed approaches deal with the boolean dataset where we only store presence or absence of an item in particular transaction. They can be modified to work in the situtations where we need to deal with non-boolean datasets. Some new heuristics apart from degree of conflict can be examined for selecting the transactions for removal of sensitive itemsets in order to further improve the utility of the data.

REFERENCES

[1] Saygin, Yücel, Vassilios S. Verykios, and Ahmed K. Elmagarmid. "Privacy preserving association rule mining." Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems, 2002. RIDE-2EC 2002. Proceedings. Twelfth International Workshop on. IEEE, 2002.

[2] Stanley R. M. Oliveira, Osmar R. Za¨ıane2, Privacy Preserving Frequent Itemset Mining, IEEE international conference on Privacy, security and data mining, pp. 43-54, 2002.

[3] V. S. Verykios, A. Elmagarmid, E. Bertino, Y. Saygin, E Dasseni, Association Rule Hiding, IEEE Transactions on Knowledge And Data Engineering, vol. 16, no. 4, pp. 434-447, 2004.

[4] P. Cheng, J. F. Roddick, S. C. Chu, C.W. Lin, Privacy preservation through a greedy, distortion-based rule-hiding method, Applied Intelligence, vol. 44, no. 2, pp. 295-306, 2015.

[5] Charu C. Agarwal, Philip S. Yu. Privacy-Preserving Data Mining Models and Algorithms. Springer ISBN: 978-0-387-70991-8 (Print) 978-0-387-70992-5 (Online)

[6] G. Lee, C. Chang, A. L.P Chen, Hiding Sensitive Patterns in Association Rules Mining, 28th Annual International Computer Software and Applications Conference, pp. 424-429, 2004.

[7] C. Lin, T. Hong, K. Yang, S. Wang, The GA-based algorithms for optimizing hiding sensitive itemsets through transaction deletion, Applied Intelligence, vol. 42, no. 2, pp. 201-230, 2015.

[8] S. R. M. Oliveira, O. R. Zaïane. Protecting sensitive knowledge by data sanitization. 3rd IEEE International Conference on Data Mining (ICDM), pages 211– 218, 2003.

[9] A. Amiri. Dare to share: Protecting sensitive knowledge with data sanitization. Decision Support Systems, 43(1):181–191, 2007.

[10] M. Berry, G. Linoff, Data Mining Techniques—for Marketing, Sales, and Customer Support, John Wiley and Sons, New York, USA, 1997

[11] Gkoulalas-Divanis, Aris, and Vassilios S. Verykios. Association rule hiding for data mining. Vol. 41. Springer Science & Business Media, 2010.

[12] Atallah, Mike, et al. "Disclosure limitation of sensitive rules." *Knowledge and Data Engineering Exchange, 1999.(KDEX'99) Proceedings. 1999 Workshop on*. IEEE, 1999.

[13] Sun, Xingzhi, and Philip S. Yu. "A border-based approach for hiding sensitive frequent itemsets." *Data Mining, Fifth IEEE International Conference on*. IEEE, 2005.

[14] Menon, Syam, Sumit Sarkar, and Shibnath Mukherjee. "Maximizing accuracy of shared databases when concealing sensitive patterns." *Information Systems Research* 16.3 (2005): 256-270.

[15] Zaki, Mohammed J. "Mining Closed & Maximal Frequent Itemsets." *NSF CAREER Award IIS-0092978, DOE Early Career Award DE-FG02-02ER25538, NSF grant EIA-0103708* (1).

LIST OF PUBLICATIONS

Himanshu Makkar, Durga Toshniwal. “Privacy Preserving Frequent Itemset Mining with Reduced Sensitive Itemsets.” *27th ACM International Conference on Information and Knowledge Management*. ACM, 2018. [Communicated]