A Dissertation Report

On

Parallelised Hiding of Sensitive Patterns for Privacy Preservation

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

of

Master of Technology

in

Computer Science and Engineering

Submitted by

Nishtha Agrawal

M.Tech II Year (17535018)

Under the guidance of

Dr. Durga Toshniwal

Professor,

Dept. of Computer Science and Engineering

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY, ROORKEE

Roorkee – 247667

May, 2019

CANDIDATE'S DECLARATION

I hereby declare that the dissertation entitled "**Parallelised Hiding of Sensitive Patterns for Privacy Preservation**" submitted by me in partial fulfilment of the requirements for the award of the Degree of Master of Technology in Computer Science and Engineering to the Department of Computer Science and Engineering, Indian Institute of Technology Roorkee is my original work carried during August 2018 to May 2019 under the guidance of Dr. Durga Toshniwal, 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 award of any other degree of this and any other institute.

Date:

Place: Roorkee Nishtha Agrawal

CERTIFICATE

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

Date:

Place: Roorkee

Dr. Durga Toshniwal

(Professor)

(Department of Computer Science and Engineering)

(Indian Institute of Technology Roorkee)

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude towards supervisor Dr. Durga Toshniwal for the continuous support, motivation, and guidance. I would also like to thank my lab-mates and friends for their support and help.

I am also grateful to the Department of Computer Science & Engineering of IIT Roorkee for providing valuable resources to aid my research.

Nishtha Agrawal

ABSTRACT

Frequent itemset mining is a field of data mining where frequent itemsets are extracted from the dataset. This may reveal some sensitive information which is not meant to be shared with third party. Privacy Preserving Data Mining approaches are used to hide that sensitive information from the dataset but along with that they also have some side effects on the datasets. Among the three types of Privacy Preserving Data Mining methods, Heuristic-based are better in terms of scalability and time efficiency as compared to the border-based and exact approaches. Heuristics-based Privacy Preserving Data Mining approaches are used to sanitize the dataset i.e., removal of sensitive patterns from the transactions, based on some heuristics. So far most of the existing techniques used for hiding sensitive patterns make use of candidate-based pattern generation methods for generating frequent patterns which takes a lot of time because a large candidate itemset space is generated. In this work, we have proposed FP-Tree based Sensitive Patterns Removal (FSR) approach. This proposed approach makes use of candidate-less pattern generation technique for hiding the sensitive patterns which reduces a lot of time as compared to previous techniques. Experiments have been performed on benchmark dataset where the proposed approach has resulted into the sanitized data with substantially better utility and better time efficiency as compared to the existing approaches. But these sequential approaches are not able to cope up with the big data. So, there is another proposed approach- Parallelised FP-Tree based Sensitive Patterns Removal (PFSR), which is the parallel implementation of Proposed FSR approach on spark parallel computing framework. This parallelised approach is scalable enough for handling large dataset. Experiments performed using benchmark datasets shows that Proposed PFSR approach scales better as compared to Proposed FSR approach, and other existing sequential approaches.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND MOTIVATION

1.1 Introduction

Frequent Pattern Mining is an important technique used by organizations in order to discover the information or useful patterns from large amount of transactional dataset for more profitable business. Along with pattern generation, there are chances that some private information also gets mined. This information is sensitive to the organization and cannot be shared with the third party. Therefore, there comes a challenge to mine the useful patterns in such a way that the sensitive or confidential information remains hidden.

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. As a result of collaborative data mining, privacy is quite essential.

Threats caused by data mining techniques can be of two types: (i) Data itself contains some private information which might be a threat and is known as *data privacy.* (ii) Some confidential information can be extracted from the knowledge mined from datasets which is known as *knowledge privacy.* Hence, Privacy Preservation Data Mining comes into the picture here. It is the field in which the confidential or sensitive information has to be hidden from the transactional datasets before releasing it to the third party for preserving its privacy.

Sensitive pattern/information comprises of some confidential or inside information of the individual or the organization such as company policies, security/identity number of an individual, bank transactions details etc., which are not meant to be shared with third party. Sensitive pattern hiding method sanitizes or hides the sensitive patterns from the knowledge extracted from the results, obtained after applying any rule or pattern mining algorithm on the transactional dataset.

Different Privacy Preserving Data Mining 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. 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.

Sensitive pattern hiding methodologies are fairly divided into three primary categories: Exact approaches, border-based approaches and heuristics-based approaches. Exact and border-based approaches are complicated to implement and really time-consuming, therefore these approaches do not suit for large datasets, because as the dataset size increases the computational time also increases exponentially. On the other-hand, Heuristics-based approaches are simpler in implementation as compared to the other two methods. Heuristics-based approaches are efficient and fast as compared to the exact and border-based approaches as they take decision based on local optima. These techniques provide good approximate solution; therefore, these techniques are of major importance for data scientists.

1.2 Problem Statement

There are many existing privacy-preserving data mining models which can be used to hide sensitive patterns from the result of data mining by transforming the data which affects the utility of the data. Along with that, these techniques are time inefficient due to large itemset space involved when run on a large dataset. So, the main motive of the present work is to create an improved technique for hiding the sensitive itemsets with better time efficiency and accuracy for large datasets. The existing approaches are not capable of handling for the case where the data is present in huge amount. So, the other objective is to design the parallel model for the process of hiding sensitive patterns which can handle the big data.

1.3 Specific Research Contributions

Following are the two Research contributions:

1. In this work, we have proposed a heuristics-based approach called FSR (FP-Tree based Sensitive Patterns Removal). FSR is the sequential approach for hiding of the sensitive patterns. This approach tries to reduce the side effects of sanitization process over the dataset and also reduce the time taken for complete sanitization by making use of candidate-less pattern generation technique. In candidate-based generation the effect of hiding victim items from k-itemset in sensitive transaction in Apriori algorithm is propagated to k+1-itemsets, which leads to more misses cost, but in candidate-less

pattern generation it does not affect all other frequent patterns. Therefore, it reduces the side effects and also candidate-less pattern generation technique does not produce a huge candidate set unlike previous techniques and hence reduces the data sanitization time.

2. Along with this, one another approach PFSR (Parallelised FP-Tree based Sensitive Patterns Removal) has also been proposed which is the parallel implementation of proposed FSR approach over spark framework. Spark is used in order to take advantage of the in-memory computational model which allows us to reduce the time required by the algorithm and hence makes the approach more scalable.

1.4 Organisation of Report

This report is organised in five different sections. The current section gives a brief introduction about the privacy preservation in data mining. Other section contains:

Section 2: This section contains what has been done so far or previous works in the field of privacy preserving data mining. It also contains the research gaps.

Section 3: This section contains the proposed solution which will improve the performance of hiding of sensitive information.

Section 4: This section contains the results of experiments done on the synthetic dataset and their comparison with previous algorithms.

Section 5: This section contains the conclusion and discuss the future work.

CHAPTER 2

RELATED WORK

2.1 Privacy Preserving Data Mining

Privacy Preserving Data Mining (PPDM) is the field which is used to hide confidential or sensitive information from the extracted knowledge from the transactional datasets for preserving its privacy. PPDM approaches also reduce the utility of the dataset.

Threats caused by data mining techniques can be of two types: (i) Data itself contains private information which can be a threat and also known as *data privacy*. (ii) Some confidential information can be extracted from the knowledge mined from datasets which is known as *knowledge privacy*.

2.2 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. In market basket analysis, a transactional Dataset D consists of many transactions and each transaction has a unique identifier TID.

Table I shows an example of general transactional dataset:

TABLE I. DATASET D

Transaction ID	Items
T1	ABC
T2	ABCD
T3	BCE
T4	ACDE
T5	DE
T6	AB

Each transaction T_i contains a set of items with it. From this transactional data organizations try to find the interesting patterns i.e. Frequent Patterns. A predefined minimum support threshold is given based upon which the frequent patterns are identified. This technique discovers all those patterns in which the support values of the itemsets are greater than the given minimum threshold denoted by σ which is provided by the organization itself.

2.2.1 Candidate-Less Pattern Generation (FP-Growth)

There are two types of pattern generation techniques: a). Candidate based pattern generation (Apriori) and b). Candidate Less pattern generation (FP-Growth). The Apriori algorithm is based on the fact that if a subset S appears k times, any other subset S' that contains S will appear k times or less. So, if S doesn't pass the minimum support threshold, neither does S'. There is no need to calculate S', it is discarded a priori. FP-Growth is an improvement of Apriori designed to eliminate some of the heavy bottlenecks in Apriori. FP-Growth [3] simplifies all the problems present in Apriori by using a structure called an FP-Tree. In an FP-Tree each node represents an item and its current count, and each branch represents a different association. In FP-Growth Algorithm, each transaction is sorted in descending order of the support of each item present in it and then each item is added to tree.

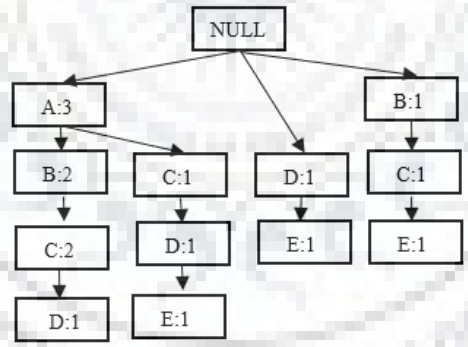

Figure 1. FP-Tree for Dataset D as shown in Table I.

For the dataset shown in Table 1, FP-Tree in figure 1 is built. Let minimum threshold i.e. σ be 2. Therefore, the frequent patterns generated from this *D* are: {ABC, ACD, CE and DE}.

2.3 Sensitive Pattern Hiding

Frequent Pattern Mining can also be a threat to privacy and information security if not done or used properly. Sensitive Pattern Hiding (SPH) is that field of data mining which provides the ways to prevent sensitive itemsets present inside the data from getting revealed. The key idea

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

2.4 Metrics for Performance Analysis

Consider *S* as set of sensitive itemset, *OF as set of frequent itemset in original database and SF as set of frequent itemset in sanitized database*. Hiding Ratio is used to check the efficiency of the proposed algorithm. More is the Hiding ratio more the proposed algorithm is efficient. Equation 1 denotes the Hiding ratio:

$$Hiding\ Ratio = SF/OF \tag{1}$$

Misses Cost denotes the number of legitimate non-sensitive patterns which got hidden after the sanitization process. Lesser is the Misses cost more the proposed algorithm is efficient. Equation 2 denotes the Misses cost:

$$Misses\ Cost = OF\text{-}SF \tag{2}$$

There are two other parameters upon which the quality of solution depends *Hiding Failure* and *Pseudo Patterns.* Hiding Failure represents the set of sensitive itemsets which are still present in the updated database after the sanitization process has been applied to the original database. Pseudo Patterns represents those patterns that were not frequent in the original database but after the application of sanitization process, they are converted to the frequent itemsets in the sanitized database.

2.5 Literature Survey

Oliveira & Zaïane [4] provided a two scans solution, in which multiple itemsets get hidden in two scans of dataset only. First scan is used to create an index file, which was used to efficiently retrieve sensitive transaction for any sensitive itemset. Second scan is used to sanitize the data

such that non-sensitive patterns are affected minimally. They introduced three algorithms: MaxFIA- Maximum Frequent Itemset Algorithm, MinFIA- Minimum Frequent Itemset Algorithm and IGA-Itemset Grouping Algorithm. MinFIA works as follows, first the sensitive transactions are identified; sensitive transactions are those transactions which contain any sensitive patterns. After that they are sorted according to the degree of conflict. Then from each transaction (depending upon the threshold), victim item is removed, victim item for each sensitive pattern is chosen as the one with maximum support. MaxFIA works in the same manner but instead of choosing the victim item as the one with maximum support, it is chosen as the one with minimum support. IGA works as follows: Common items in sensitive itemsets are grouped together and then victim item is the one which is having minimum support and is shared by all the itemsets of that group.

Verykios [6] proposed a confidence-based approach. According to this approach the sensitive patterns are hidden by decreasing the confidence of an association rule because this causes lesser side effects to the sanitized dataset. But this approach does not any guarantee hiding of all the sensitive patterns.

Cheng [7] introduced another heuristic approach. According to this approach, in first step for each transaction store a count of non-sensitive patterns it supports. In second step for each sensitive pattern store count of transaction it supports. Then transaction identified in the second step are sorted according to their count calculated in first step. Then to the threshold, it removes the victim itemset from the transactions. Victim item is the item in sensitive pattern which has maximum support.

Oliveira & Zaïane [5] proposed one another approach, called SWA (Sliding Window Algorithm). In this approach, first all the transactions that does not support any sensitive patterns are copied to sanitize database and then for each sensitive pattern we select the victim item with the maximum support, and then based on threshold it is removed from group of remaining transaction.

A. Amiri [8] proposed three approaches: Aggregate approach: in this approach the transaction which is removed from the dataset is chosen in the following way- the transaction which supports a smaller number of non-sensitive frequent patterns but a large number of sensitive patterns. Disaggregate approach: In this approach the item is removed from the transaction rather than whole transaction. The victim item is chosen in the same manner as in above

method the transaction is chosen. Hybrid approach: in this approach, the transaction is chosen according to the aggregate approach and the item to be removed is chosen according to the disaggregate method.

The techniques used for hiding sensitive patterns in all the above discussed techniques are based on candidate-based pattern generation which takes a lot of computational time, therefore using candidate less approach for the same can drastically reduce the time.

2.6 Research Gaps Identified

From the detailed survey of previous work, these are some research gaps that have been identified:

Heuristics-based methods are better than border-based and exact approaches for sanitization. Thus, there is a scope of better heuristics that can be used for hiding the sensitive patterns in order to preserve the utility and is time efficient as well.

There is a scope of using non candidate-based generation techniques so as to save the candidate set generation process and thus a lot of time can be saved during sanitization.

Most of the work done so far for hiding the sensitive patterns is sequential. Thus, there is need for Parallelisation of sensitive pattern hiding technique which will reduce the computational time.

CHAPTER 3

PROPOSED FRAMEWORK

A FP-Tree based Sensitive Patterns Removal (FSR) approach has been proposed. The proposed approach uses the advantage of candidate-less pattern generation technique. i.e. FP Tree.

3.1 Proposed FSR Approach

In figure 2, the framework of Data sanitization process is described. Original dataset and sensitive patterns are provided as input to the algorithm. Sanitized dataset is obtained as the output of the algorithm. Data Sanitization using FP-Tree is the block where the proposed approach works.

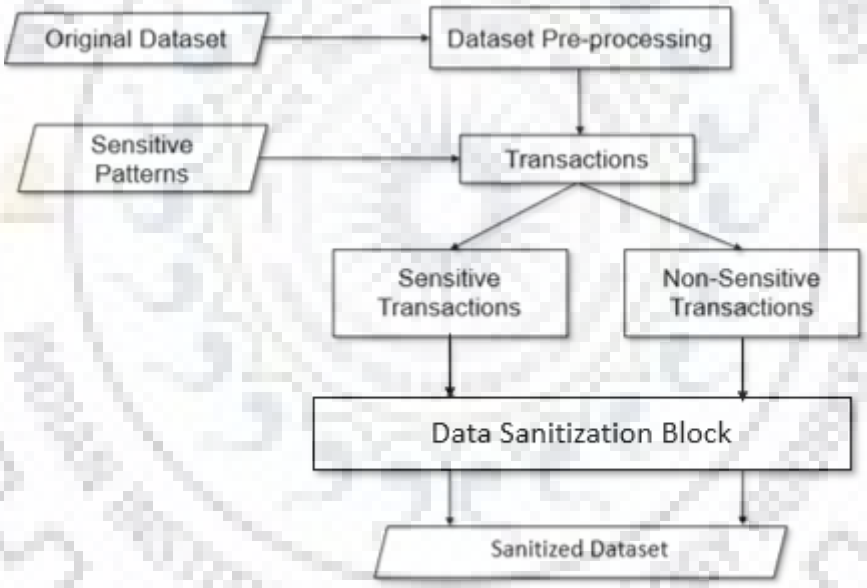

Figure 2. Data Sanitization Framework.

In this proposed FSR (FP-Tree based Sensitive Patterns Removal) approach, the transactions are divided into different sets: Sensitive Transactions and Non-Sensitive Transactions. Sensitive transactions are those transactions which contains any sensitive pattern as whole or any subset in itself. Two sets are created one with sensitive transactions which is gone for further sanitization process and the non-sensitive transactions are simply added to sanitized dataset. In the same scan the support of each item of transactional data is also calculated.

Data is sanitized using victim item and victim item is chosen on the basis of its support (the one with minimum support) in its respective itemset.

The proposed approach will have 0% hiding failure, as all the sensitive patterns gets hidden i.e., no frequent pattern will be generated from the sanitized data that would contain any sensitive pattern. And also, zero pseudo patterns will be generated i.e., patterns which were not present in original dataset will also not be generated from the sanitized dataset. But there will be some misses cost i.e., some legitimate non-sensitive patterns may get hidden after the sanitization process.

3.2 Proposed PFSR Approach

Parallelised FP-Tree based Sensitive Pattern Removal (PFSR) is the parallelized version of FSR. This approach is implemented over Apache Spark distributive framework. This approach is designed to run in the distributed environment (Hadoop File System or Standalone Spark) over multiple nodes for parallel processing.

Parallel execution over multiples nodes reduces the time taken by sanitization process. The dataset is divided into multiple sub datasets and each of the sub dataset is used and handled parallelly. Algorithm for the same has been divided into four steps:

1. Data Pre-Processing
2. Identifying Sensitive Transactions.
3. Data Sanitization
4. Aggregation

DATA PRE-PROCESSING

In this step the support-count of each and every item is computed and all the transactions that are present in the dataset are divided non-overlapping groups.

IDENTIFYING SENSITIVE TRANSACTIONS

For each transaction in the group whether any of the sensitive itemset is its sensitive or not is checked. If it is not a subset of any sensitive pattern then mark that transaction as non-sensitive transaction. But if it is a subset then that transaction is sensitive transaction.

DATA SANITIZATION

Data is sanitized using victim item of sensitive transactions and victim item is chosen on the basis of its support (the one with minimum support) in its respective itemset (sensitive transaction.

AGGREGATION:

As each of the group is independent of other group, therefore aggregation of these FP-Trees is done. Recursively all the FP-Trees are combined to form one global FP-Tree.

Like FSR the proposed PFSR approach will have 0% hiding failure, as all the sensitive patterns gets hidden i.e., no frequent pattern will be generated from the sanitized data that would contain any sensitive pattern. And also, zero pseudo patterns will be generated i.e., patterns which were not present in original dataset will also not be generated from the sanitized dataset. But there will be some misses cost i.e., some legitimate non-sensitive patterns may get hidden after the sanitization process.

CHAPTER 4

EXPERIMENTS AND DISCUSSION

Several experiments have been conducted to measure the efficiency of the proposed algorithm. All the experiments were conducted on the single Ubuntu workstation having 48 cores, 64GB memory, running Hadoop version 2.7.6 with spark version 2.4.0 in a standalone mode. The performance of the proposed hiding algorithm has been analysed by comparing it with the earlier approaches.

4.1 Datasets Used

The dataset used for the experiments was generated by the IBM Synthetic data generator, which is a standard tool for this type of dataset. Different datasets are generated for the analysis purpose varying in number of transactions and number of sensitive patterns.

TABLE II. DATASET DESCRIPTION

Number of Transactions	Average Transaction Length	Min. Support (%)					Number of sensitive patterns				
10,000	10	5	8	10	12	15	3	10	20	50	100
20,000	15	5	8	10	12	15	3	10	20	50	100
50,000	20	5	8	10	12	15	3	10	20	50	100
100,000	30	5	8	10	12	15	3	10	20	50	100
10,00,000	40	5	8	10	12	15	3	10	20	50	100

4.2 Experimental Results

Various types of results have been generated by comparing proposed FSR approach and MinFIA. Hiding Ratio, Misses cost and Running cost of Proposed FSR has been compared with traditional MinFIA Algorithm. All three metrics has been calculated for three different parameters - Number of transactions, Number of sensitive itemsets and minimum support. Experiments are conducted by varying one parameter and while keeping other two constant. All the experiments have been conducted several times and therefore average value of all those experiments have been shown in graphs below.

Proposed PFSR Approach is the parallelised version of FSR hence it performs in similar manner as Proposed FSR in terms of Hiding Ratio and Misses Cost. Hence only running time is analysed for Proposed PFSR.

4.2.1 Analysis of Hiding Ratio

In this case, minimum support (%) keeps varying i.e., n = 5, 8, 10, 12 and 15 while the number of transactions is kept constant to 100000 and the number of sensitive itemsets are kept constant to 100.

It can be observed from the above figure 3, that the hiding ratio of the proposed FSR is better as compared to that of MinFIA. There is gain of around 4% in hiding sensitive patterns for proposed FSR Approach.

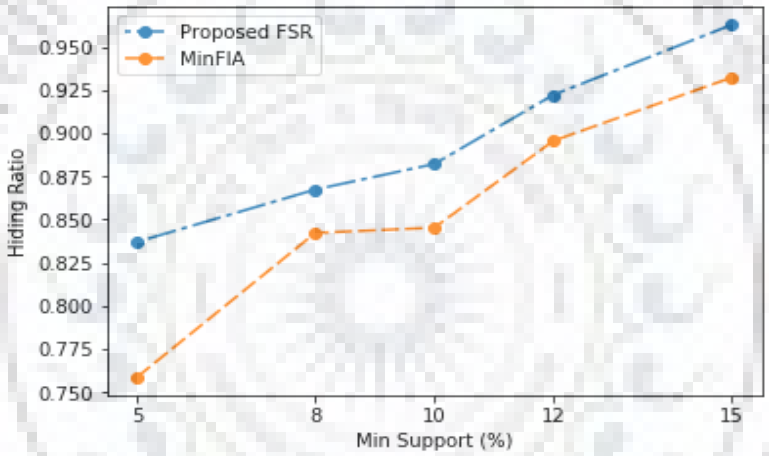

Figure 3. Hiding Ratio of Proposed FSR vs MinFIA for varying minimum support where T = 100,000 and NS = 100.

In this case, the number of transactions keeps varying i.e., n = 10000, 20000, 50000 and 100000 while the number of sensitive itemsets are kept constant to 100 and minimum support threshold is kept constant to 10%.

It can be observed from the above figure 4, that the hiding ratio of the proposed FSR is better as compared to that of MinFIA. There is gain of around 6% in hiding sensitive patterns for proposed FSR Approach. The proposed FSR approach also has 0% hidden failure and 0% artificial patterns.

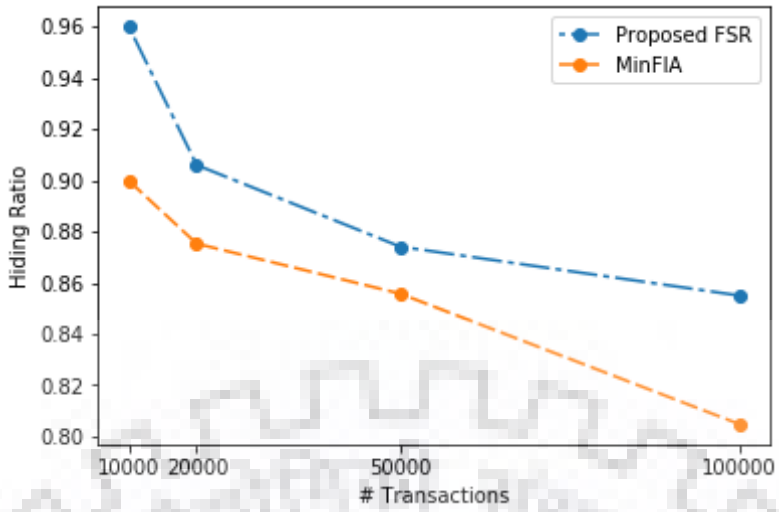

Figure 4. Hiding Ratio of Proposed FSR vs MinFIA for varying number of transactions where NS = 100 and MS = 10%.

In this case, the number of sensitive itemsets keeps varying i.e., n = 3, 10, 20, 50 and 100 while the number of transactions is kept constant to 100000 and minimum support threshold is kept constant to 10%.

It can be observed from the above figure 5, that the hiding ratio of the proposed FSR is better as compared to that of MinFIA. There is gain of around 2.5 % in hiding sensitive patterns for proposed FSR Approach.

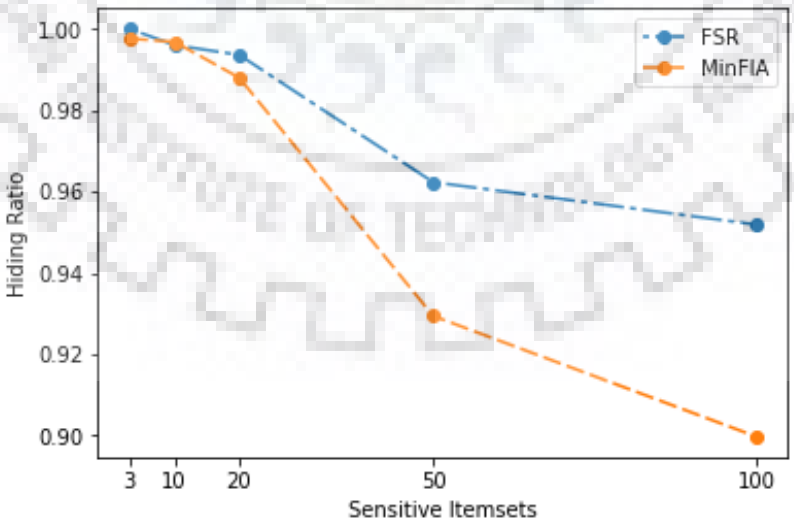

Figure 5. Hiding Ratio of Proposed FSR vs MinFIA for varying number of sensitive itemsets where T =100,000 and MS = 10%.

4.2.2 Analysis of Misses Cost

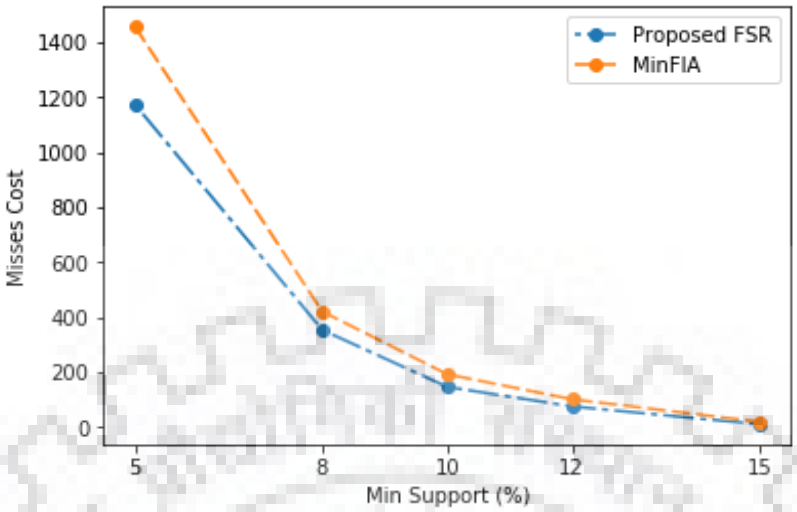

Figure 6. Misses cost of Proposed FSR vs MinFIA for varying minimum support where T= 100,000 and NS = 100.

In this case, minimum support (%) keeps varying i.e., n = 5, 8, 10, 12 and 15 while the number of transactions is kept constant to 100000 and the number of sensitive itemsets are kept constant to 100. It can be observed from the above figure 6, that the misses cost of the proposed FSR is lower as compared to that of MinFIA.

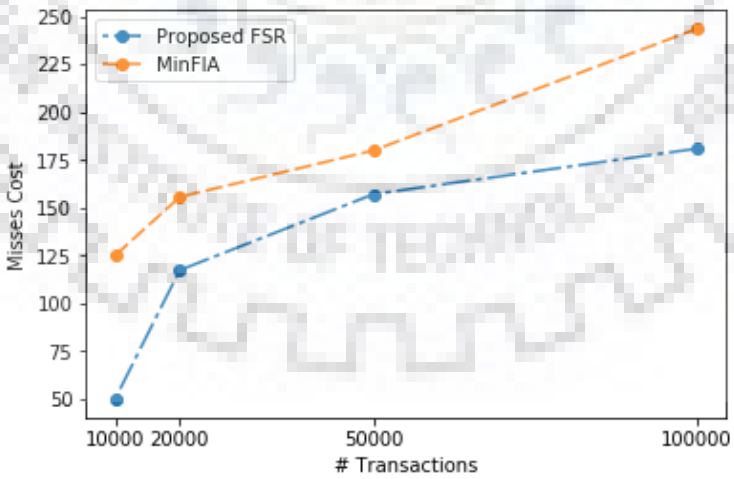

Figure 7. Misses cost of Proposed FSR vs MinFIA for varying number of transactions where NS = 100 and MS =10%.

In this case, the number of transactions keeps varying i.e., n = 10000, 20000, 50000 and 100000 while the number of sensitive itemsets are kept constant to 100 and minimum support threshold is kept constant to 10%.

It can be observed from the above figure 7, that the misses cost of the proposed FSR is lower as compared to that of MinFIA. There is gain of around 50 patterns i.e., around 50 non-sensitive frequent patterns were getting hidden by MinFIA.

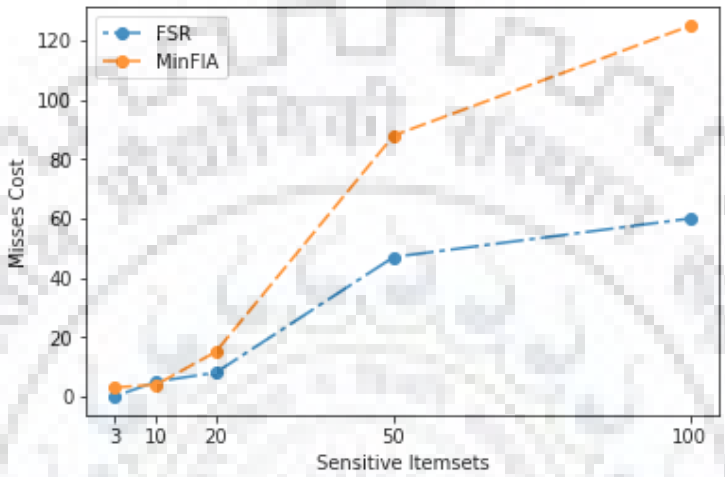

Figure 8. Misses cost of Proposed FSR vs MinFIA for varying number of sensitive itemsets where T = 100,000 and MS = 10%.

In this case, the number of sensitive itemsets keeps varying i.e., n = 3, 10, 20, 50 and 100 while the number of transactions is kept constant to 100000 and minimum support threshold is kept constant to 10%.

It can be observed from the above figure 8, that the misses cost of the proposed FSR is lower as compared to that of MinFIA. There is gain of around 30 to 40 patterns i.e., around 30 to 40 non-sensitive frequent patterns were getting hidden by MinFIA.

4.2.3 Analysis of Running Time

In this case, minimum support (%) keeps varying i.e., n = 5, 8, 10, 12 and 15 while the number of transactions is kept constant to 10000 and the number of sensitive itemsets are kept constant to 100.It can be observed from the above figure 9, that the running time of the proposed FSR is faster as compared to that of MinFIA.

In this case, the number of transactions keeps varying i.e., n = 10000, 20000, 50000 and 100000 while the number of sensitive itemsets are kept constant to 100 and minimum support threshold is kept constant to 10%.

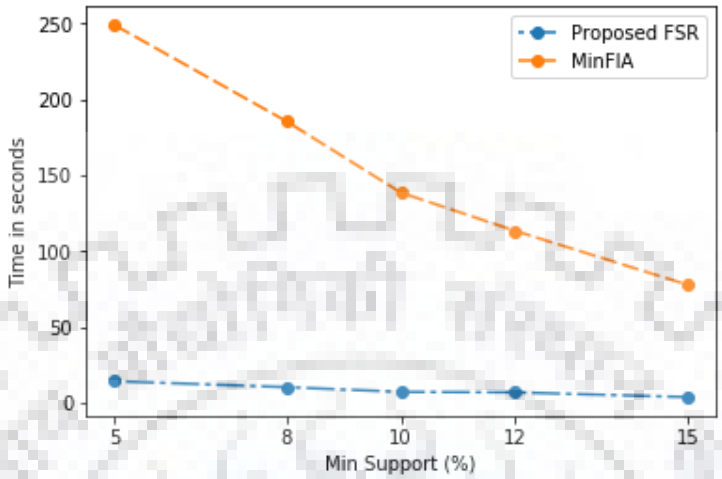

Figure 9. Running Time of Proposed FSR vs MinFIA for varying minimum support where NS = 100 and T = 10000.

It can be observed in figure 10, with the increase in number of transactions, for MinFIA a large number of candidate space will be generated which take huge amount of time. While for Proposed FSR approach only few seconds are taken for the data sanitization.

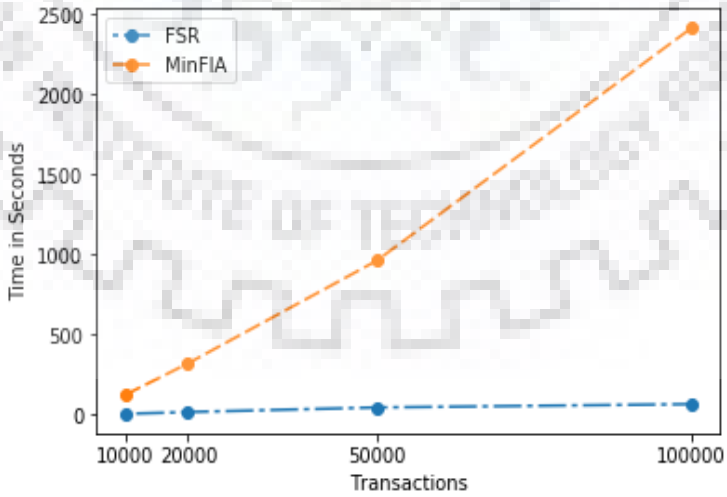

Figure 10. Running Time of Proposed FSR vs MinFIA for varying number of transactions where MS = 10% and NS = 100.

In this case, the number of sensitive itemsets keeps varying i.e., n = 3, 10, 20, 50 and 100 while the number of transactions is kept constant to 10000 and minimum support threshold is kept constant to 10%.

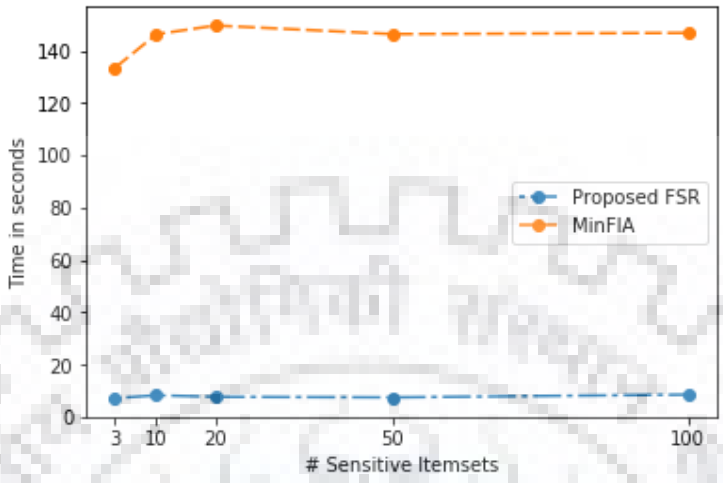

Figure 11. Running Time of Proposed FSR vs MinFIA for varying number of sensitive itemsets where T = 10000 and MS = 10%.

4.2.4 Analysis of Running Time of Proposed PFSR vs Proposed FSR

Figure 12 shows the running time of proposed PFSR approach on Benchmark dataset (10,000 number of transactions) and compares it with the running time of sequential FSR approach and MinFIA approach.

Minimum support threshold was set to 10% for conducting this set of experiments. It can be observed that MinFIA takes a lot of time as compared to Proposed FSR and Proposed PFSR approaches. Proposed PFSR performs better than both Proposed FSR and MinFIA but still it is relatively comparable with FSR (again shown in figure 13) but it totally outcasts MinFIA approach.

Proposed PFSR is the parallel implementation of proposed FSR on Spark Framework. It is equally efficient as FSR in terms of these performance metrics - hidden ratio and misses cost. The parallel implementation on spark reduces the time taken SPH approach. Traditional approaches were sequential and take time when operated on large dataset as processing the large dataset 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.

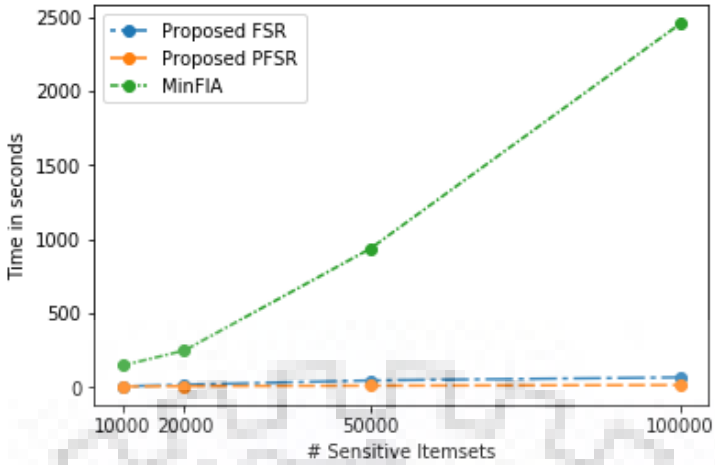

Figure 12. Running Time of Proposed PFSR vs Proposed FSR vs MinFIA for varying number of transactions where MS = 10% and NS = 100.

In Figure 13, running time comparison of proposed FSR(sequential version) has been done with Proposed Parallelised FSR. The comparison has been done for different number of transactions.

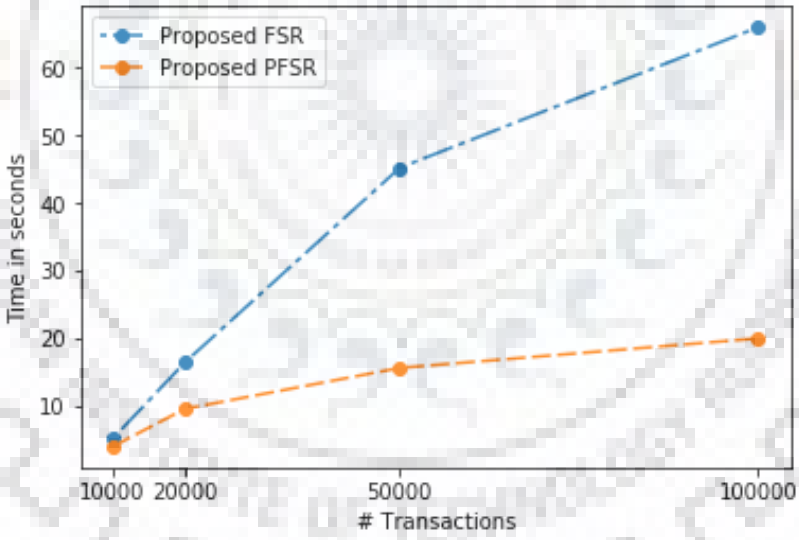

Figure 13. Running time Comparison of Proposed FSR vs Proposed PFSR for varying number of transactions where NS = 100 and MS = 10%.

Figure 14 shows the running time of proposed PFSR approach on Benchmark dataset (100,000 number of transactions) and compares it with the running time of sequential FSR approach and MinFIA approach. It can be observed that MinFIA takes a lot of time as compared to Proposed FSR and Proposed PFSR approaches. Proposed PFSR performs better than both Proposed FSR

and MinFIA but still it is relatively comparable with FSR (again shown in figure 15) but it totally outcasts MinFIA approach.

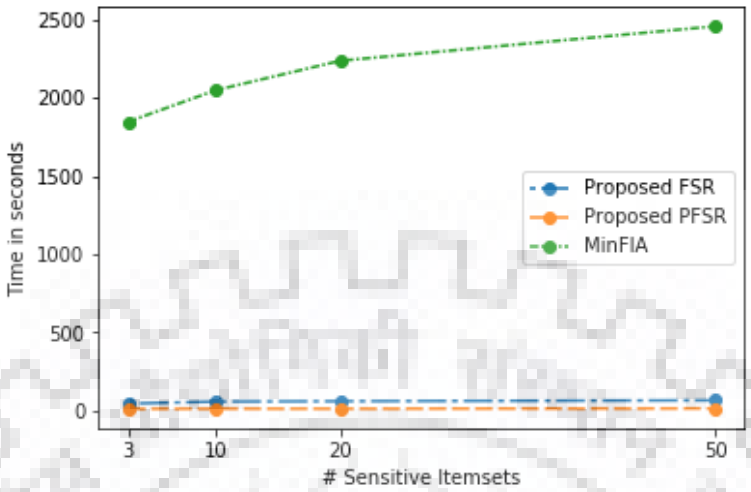

Figure 14. Running time of Proposed FSR vs Proposed PFSR vs MinFIA for T = 100,000 and MS = 10%

Figure 15 shows the running time of proposed PFSR approach on Benchmark dataset (100,000 number of transactions) and compares it with the running time of sequential FSR approach. Minimum support threshold was set to 10% 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 3 to 50.

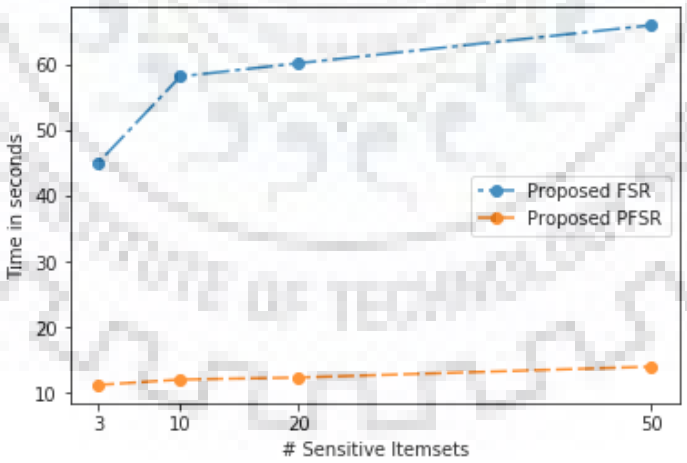

Figure 15. Running time Comparison of Proposed FSR vs Proposed PFSR for T = 100,000 and MS = 10%

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. Proposed PFSR took 30-40 sec(approx.) lesser than the Proposed FSR approach. It can also be concluded from

the results that running time of both the approaches increases with the increase of number of sensitive itemsets.

To analyse the performance of proposed PFSR in better way, dataset size was further increased and same set of experiments were conducted on benchmark dataset (1,000,000 number of transactions). It can be concluded from the Figure 16 sequential approach FSR did not scale well when number of sensitive itemsets were increased.

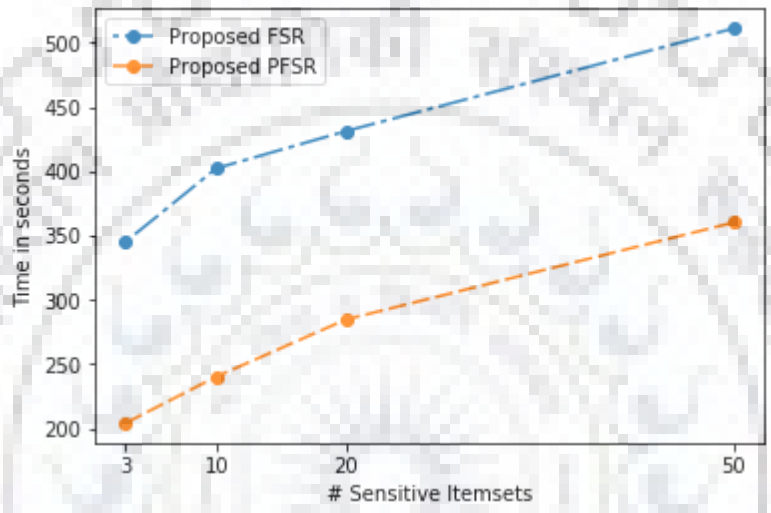

Figure 16. Running time Comparison of Proposed FSR vs Proposed PFSR for T = 1,000,000 and MS = 10%

Proposed PFSR was able to scale well on large dataset whereas proposed FSR approach did not scale well and took relatively very large time. It can be concluded that with the increase of the size of dataset, proposed PFSR is able to scale well but sequential approaches- Proposed FSR and MinFIA is not able to scale well.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Out of Border-based, Exact and Heuristics-based approaches, we focused on heuristics-based because heuristic-based approaches gain advantage in the field of memory-efficiency and scalability, although they also cause side-effects.

In this work, proposed approach FSR (FP-Tree based Sensitive Pattern Removal) makes use of candidate-less pattern generation technique for hiding the sensitive patterns. This helps in overcoming the disadvantage of previous techniques (which are relied on candidate-based pattern generation) which are large candidate itemset and time inefficiency. Proposed approach reduces the time efficiently and it also provides better accuracy of results as it has more hiding ratio and lesser misses cost as compared to current state-of-the-art algorithm MinFIA. Therefore, proposed approach is suitable for large datasets because it takes very less time as compared to earlier approaches. The proposed approach is tested extensively by performing different experiments under varying parameters. The experiments were performed on benchmark dataset. Proposed approach has performed relatively better with respect to traditional approach.

Parallelized approach- PFSR (Parallelised FP-Tree based Sensitive Pattern Removal) has been proposed further which parallelizes the work of FSR on spark framework. This approach scales well with massive amount of data if proper resources are available for the implementation of spark cluster. Experiments were performed on different large benchmark datasets where the parallelized proposed approaches scaled greatly with increased load.

5.2 Future Work

The experiments were conducted on benchmark IBM synthetic dataset, but in future the proposed approach can also be made to work on real life organisational data as well. There is also scope of a better-heuristics while creating the non-over lapping sets from transaction data in Proposed PFSR for further improvement in its time efficiency.

REFERENCES

[1] Agrawal R., Srikant R. Privacy Preserving Data Mining. ACM SIGMOD, *International Conference on Management of data*, 2000.

[2] R. Agrawal, A. Arning, T. Bollinger, M. Mehta, J. Shafer, R. Srikant, "The Quest data mining system", *Proceedings of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining*, Aug. 1996.

[3] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In W. Chen, J. Naughton, and P. A. Bernstein, editors, 2000 *ACM SIGMOD Intl. Conference on Management of Data*. ACM Press, May 2000.

[4] Stanley R. M. Oliveira, Osmar R. Zaıane2, Privacy Preserving Frequent Itemset Mining, *IEEE international conference on Privacy, security and data mining*, pp. 43-54, 2002.

[5] 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.

[6] 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.

[7] G. Lee, C. Cheng, A. L.P Chen, Hiding Sensitive Patterns in Association Rules Mining, *28thAnnual International Computer Software and Applications Conference*, pp. 424-429, 2004.

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

[9] Charu C. Agarwal, Philip S. Yu. Privacy-Preserving Data Mining Models and Algorithms. Springer ISBN: 978-0-387-70991-8 (Print) 978-0-38770992-5 (Online)

[10] 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.

[11] 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, 2016.

[12] Haoyuan Li, Yi Wang, Dong Zhang, Ming Zhang, Edward Y. Chang, "PFP: Parallel FP-Growth for Query Recommendation", *Proceedings of the 2008 ACM conference on Recommender systems*, 2008.

LIST OF PUBLICATIONS

[1]. Nishtha Agrawal and Durga Toshniwal, "Improved Hiding of Business Sensitive Patterns using candidate-less Approach", The 18th Int'l Conf on Information and Knowledge Engineering (IKE 2019), (Qualis - B4, ERA - C) [Accepted]

[2]. Nishtha Agrawal and Durga Toshniwal, "Parallelised Hiding of Business Sensitive Patterns using candidate-less Approach", in 28th International Joint Conference on Artificial Intelligence (IJCAI 2019), The 1st Workshop on Artificial Intelligence for Business Security (AIBS), (Qualis – A1, ERA - A) [Communicated]

Parallelised Hiding of Sensitive Patterns for Privacy Preservation

ORIGINALITY REPORT

3%	3%	2%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.singularities.com Internet Source	2%
2	core.ac.uk Internet Source	1%
3	"Frequent Pattern Mining", Springer Nature, 2014 Publication	1%
4	airccj.org Internet Source	<1%

Exclude quotes	On	Exclude matches	< 20 words
Exclude bibliography	Off		