DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PRIVACY PRESERVING DATA ANALYSIS FOR INFORMATION SYSTEMS

by
Barış YILDIZ

October, 2022
İZMİR

PRIVACY PRESERVING DATA ANALYSIS FOR INFORMATION SYSTEM

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Engineering

by
Barış YILDIZ

October, 2022
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**PRIVACY PRESERVING DATA ANALYSIS FOR INFORMATION SYSTEMS**" completed by **BARIŞ YILDIZ** under supervision of **PROF.DR. ALP KUT** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy in Computer Engineering.

Prof. Dr. Alp KUT
Supervisor

Assoc. Prof. Dr. Derya BİRANT
Thesis Comittee Member

Asst. Prof. Dr. Gülden KÖKTÜRK
Thesis Comittee Member

Prof. Dr. Onur DEMİRÖRS
Examining Comittee Member

Assoc. Prof. Dr. Tuğba ÖZAÇAR ÖZTÜRK
Examining Comittee Member

Prof. Dr. Okan FISTIKOĞLU
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisor, Prof. Dr. Alp Kut, for his supervision and to thesis comity members Assoc. Prof. Dr. Derya BİRANT and Asst. Prof. Dr. Gulden KÖKTÜRK their guidance and encouragement.

I would like to thank TÜBİTAK-BİDEB for doctorate scholarship during my study.

I would like to thank my colleagues in Yaşar University for such a friendly environment.

I would also like to state my special thanks to my fiancé, Hatice.

Finally, I would like to state my greatest thanks to my family, my brother Özgür and my mother Fatma for always supporting me. My last words are for my father Ömer. I wish we had time for celebration. Rest in peace.

Barış YILDIZ

PRIVACY PRESERVING DATA ANALYSIS FOR INFORMATION SYSTEMS

ABSTRACT

Data collection and processing progress made data mining a popular tool among organizations in the last decades. Sharing information between companies could make this tool more beneficial for each party. However, there is a risk of sensitive knowledge disclosure. Shared data should be modified in such a way that sensitive relationships would be hidden. Since the discovery of frequent itemsets is one of the most effective data mining tools that firms use, privacy-preserving techniques are necessary for continuing frequent itemset mining. There are two types of approaches in the algorithmic nature: heuristic and exact. This study presents an exact itemset hiding approach, which uses constraints for a better solution in terms of side effects and minimum distortion on the database. The proposed approach does not require frequent itemset mining executed prior to the hiding process. This gives our approach an advantage in total running time. We give an evaluation of our algorithm on some benchmark datasets. Our results show the effectiveness of our hiding approach and elimination of prior mining of itemsets is time efficient.

In addition, we conducted a survey to understand the awareness of people regarding the sensitivity of their personal data. The results show that participants tend to protect their privacy whenever possible and have a different attitude of sensitivity in different situations. In addition, it has been observed that participants tend to give misleading information when they do not feel comfortable. This study shows that people are uncomfortable with sharing sensitive information with third parties rather than collecting it.

Keywords: frequent itemset mining, privacy preserving data mining, personal data

BİLGİ SİSTEMLERİ İÇİN GİZLİLİĞİ KORUYAN VERİ ANALİZİ

ÖZ

Veri toplama ve işlemedeki ilerleme, veri madenciliğini son yıllarda kuruluşlar arasında popüler bir araç haline getirmiştir. Şirketler arasında bilgi paylaşımı, bu aracı her bir taraf için daha faydalı hale getirebilir. Ancak, hassas bilgilerin ifşa edilmesi riski vardır. Paylaşılan veriler, hassas ilişkilerin gizleneceği şekilde değiştirilmelidir. Sık öğe kümelerinin keşfi, firmaların kullandığı en etkili veri madenciliği araçlarından biri olduğundan, sık öğe kümesi madenciliğine devam etmek için gizliliği koruyan teknikler gereklidir. Algoritmik olarak iki tür yaklaşım vardır: sezgisel ve kesin. Bu çalışma, veritabanında yan etkiler ve minimum bozulma açısından daha iyi bir çözüm için kısıtları kullanan kesin bir öğe kümesi gizleme yaklaşımı sunar. Yaklaşımımız gizleme işleminden önce sık sık öğe kümesi madenciliği yapılmasını gerektirmez. Bu, yaklaşımımıza toplam çalışma süresinde bir avantaj sağlar. Sonuçlarımız, gizleme yaklaşımımızın etkinliğini ve öğe kümelerinin önceki madenciliğinin ortadan kaldırılmasının zaman açısından verimli olduğunu göstermektedir.

Ayrıca, kişilerin kişisel verilerinin hassasiyeti konusundaki farkındalıklarını anlamak için bir anket gerçekleştirdik. Sonuçlar, katılımcıların mümkün olduğunca mahremiyetlerini koruma eğiliminde olduklarını ve farklı durumlarda farklı bir duyarlılık tutumuna sahip olduklarını göstermektedir. Ayrıca katılımcıların kendilerini rahat hissetmediklerinde yanıltıcı bilgi verme eğiliminde oldukları gözlemlenmiştir. Bu çalışma, insanların hassas bilgilerinin toplanılmasından çok üçüncü taraflarla paylaşmaktan rahatsız olduklarını göstermektedir.

Anahtar kelimeler: sık küme madenciliği, gizliliği koruyan veri madenciliği, kişisel veri

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF SYMBOLS

D	: Dataset
$\sim D$	: Intermediate form of dataset
D^s	: Sanitized dataset
T_i	:i-th transaction of the dataset
I	: Set of items
$\sigma(X)$	: Support count of itemset X in the dataset
$\sigma^s(X)$	: Support count of itemset X in the sanitized dataset
σ_{min}	: Minimum support count threshold
S	: Set of sensitive itemsets
Ss	: Set of supersets of sensitive itemsets
F	: Set of frequent itemsets in the dataset
F^n	: Set of non-sensitive frequent itemsets in the dataset
F^s	: Set of frequent itemsets in the sanitized dataset
d_{ij}	: Item of the dataset in bitmap notation at i-th row j-th column
$\sim d_{ij}$	: Item of intermediate form of the dataset in bitmap notation at i-th row j-th column
d_{ij}^s	: Item of the sanitized dataset in bitmap notation at i-th row j-th column
u_{ij}	: Binary variable of intermediate form of the dataset in bitmap notation at i-th row j-th column
SI	: Set of sibling itemsets
$SI(X)$	: Set of sibling itemsets of itemset X
r_Y	: Binary variable of constraint defined for itemset Y

ABBREVIATIONS

PPDM	: Privacy Preserving Data Mining
LHS	: Left Hand Side
RHS	: Right Hand Side
HISB	: Hiding Itemsets Using Sibling Itemset Constraints
HISB+S	: Hiding Itemsets Using Sibling Itemset Constraints with Subsets
IPA	: Integer Programming Approach
CSP	: Constraint Satisfaction Problem
NP	: Non-deterministic Polynomial-time

CHAPTER 1

INTRODUCTION

Data mining is a successful tool for extracting knowledge from large amounts of data. It is efficiently applied to many fields, such as weather forecasting (Feng & Tian, 2021), biomedical (Raja et al., 2017), medical diagnosis (Neto et al., 2019), marketing (Hong & Park, 2019), security (Amanowicz & Jankowski, 2021), and fraud detection (Sánchez-Aguayo et al., 2022). On the other hand, sensitive data used in data mining applications or sensitive knowledge gained from these applications may cause privacy breaches directly or through linkable private data. However, underlying knowledge that can be extracted using sensitive data may be valuable. Privacy-preserving data mining (PPDM) originates from the necessity to carry on performing data mining efficiently meanwhile preserve sensitive data or sensitive knowledge. Privacy protection in data mining is divided into input privacy and output privacy, which is simply depicted in Figure 1.1.

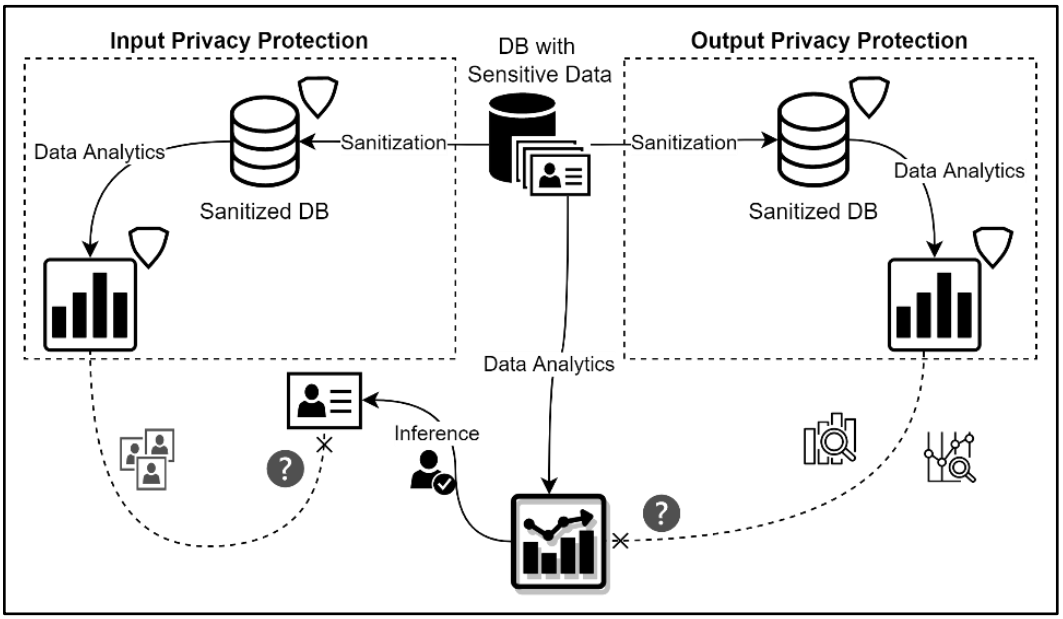


Figure 1.1 Privacy protection in data mining

Input privacy techniques aim to protect sensitive data private with such data modifications that it cannot be concluded by the outcomes of the data mining

algorithm. Achieving this requires some special techniques, including anonymization, distortion, randomization, and encryption (Liu & Özsu, 2018). Output privacy techniques aim to protect the privacy of sensitive rules or patterns and transform data in a way that all these are concealed while the remaining ones can still be revealed (Mendes & Vilela, 2017; Zhang et al., 2019).

Finding frequently co-occurring items using data mining is popular among companies to discover valuable knowledge, such as customer habits. Although this is very valuable alone, companies may be willing to share data for collaboration. In this way, a better understanding of discovered knowledge can be gained, which will help to make better strategies. However, the risk of disclosing sensitive relationships may increase. Such a scenario is given in (Verykios et al., 2004). For example, let us consider a scenario in which a supermarket sells products of two rival companies. To collaborate and increase profits, one company offers lower prices to the supermarket. The collaborator company reveals relationships with its rival's products through data mining. Using this knowledge and campaigns, the collaborator company may monopolize certain products, which can negatively affect the rival company and the supermarket. For similar situations, the stakeholders should sanitize the databases before sharing.

Technological developments have transformed individuals into data producers. Shared data between parties may contain sensitive data or knowledge about individuals. Therefore, the community is growing reactions with privacy concerns. In order to keep the trust of individuals, the development and application of privacy preserving techniques became compulsory. Otherwise, they may be less willing to share information or share misleading information, which will have a negative effect on data analytics.

1.1 Thesis Aim and Objectives

Privacy is an important aspect of data mining. Sometimes not the data but the results of data mining techniques may violate privacy. Frequent itemset mining is one of the

most effectively used data mining tools, and resulting patterns may contain sensitive knowledge. Therefore, frequent itemset hiding techniques are proposed. These approaches aim to modify the database so that sensitive itemsets or association rules are hidden and non-sensitive ones are affected minimally. Depending on their algorithmic nature, approaches may be heuristic and exact (Gkoulalas-Divanis & Verykios, 2010). Algorithms using heuristics suffer from side effects since the control of modifications on the database is limited. These algorithms are known to be faster in terms of runtime. Exact algorithms use constraints for the decision of modifications on the database. More control on modifications makes these approaches better in terms of side effects. The downside of more control and computation is consuming more time. The main objective of exact itemset hiding studies is to have fewer side effects, but these studies comprise runtime. In the case of exact approaches, there may be three ways to consume less time: having fewer constraints, skipping prior mining of the dataset for constraint generation, and decreasing the time consumption of the constraint solver. The study in this thesis focus on increasing the efficiency level without compromising the privacy quality level and data quality level.

The first main objective of this thesis is to study frequent itemset hiding and propose an approach where:

- Privacy level is preserved, and all sensitive itemsets are hidden,
- Data quality level is preserved, and side effects on non-sensitive itemsets are minimized,
- Efficiency level is increased by using fewer constraints for lessening runtime,
- Efficiency level is increased by skipping prior mining of frequent itemsets for lessening total runtime,
- Efficiency level is increased by decreasing constraint solver time consumption using relaxation techniques where the exact solution is not feasible.

The second main objective is to study the need for privacy preserving techniques. Following the technical part, a survey is carried out to understand the privacy

awareness of people and how they behave in different situations. This social part completes the technical part of the thesis.

1.2 Organization of Thesis

The organization of this thesis is given as follows:

- Section 2 gives a literature review about frequent itemset hiding.
- Section 3 gives background information. It starts with an introduction to data mining and techniques; privacy preserving data mining is introduced with some techniques.
- Section 4 introduces the proposed itemset hiding approach with evaluation. Beginning with a formal definition of itemset hiding, using constraints is presented systematically. Following that, the proposed approach is given. After a simple illustration, a comparison with a well-known approach is given on benchmark datasets.
- Section 5 introduces our survey on the awareness of people on the sensitivity of their personal data. Method and findings for frequent itemset mining and clustering are given.
- Section 6 concludes the thesis with a summary of the findings of both the technical and social parts. Lastly, possible directions for the study are given in future work.

CHAPTER 2

LITERATURE REVIEW

The first study on hiding frequent itemsets (patterns) is by (Atallah et al., 1999). The authors showed that an optimal solution to this problem is NP-hard. They propose a greedy heuristic approach that traverses the frequent itemset lattice for pinpointing the transactions and the items that they had to change so that the support of a sensitive frequent pattern reduces and falls below the support threshold. Many studies have been done after this starting point. Some approaches focus on itemset hiding, some on association rule hiding, and some propose a solution for both. Since the problem is NP-hard, there are approaches that rely on some assumptions, namely heuristic approaches.

In (Dasseni et al., 2001), the authors focus on the sanitization of sensitive rules. They reduce the confidence of these rules below the minimum confidence threshold. This approach is prone to produce ghost rules and has restrictions of hiding one rule at a time.

The work in (Oliveira & Zaïane, 2002) proposed algorithms scanning database to obtain an inverted index for transactions with sensitive items. Depending on the number of sensitive transactions that will be altered to restrict sensitive patterns, the impact on non-sensitive patterns is also calculated. Another important contribution is the metrics presented: hiding failure, misses cost, and artifactual patterns.

The authors in (Guanling Lee et al., 2004) represent the database as a binary matrix and construct a sanitization matrix consisting of values 1,0 or -1 depending on the relation between sensitive itemsets. These two matrices go through a defined multiplication process, and sanitized database is calculated in binary form.

In (Verykios et al., 2004) the hiding strategies proposed depend on finding transactions that fully or partially support the generating itemsets of a rule. The first

bunch of algorithms decreases the confidence of the rule. The second bunch of algorithms decreases the support of the rule. In order to achieve this, transactions are altered by deleting items or adding new items depending on the hiding strategy.

One interesting approach is proposed in (Saygin et al., 2001). Instead of deleting or adding items for modification on the database, authors introduce unknown values to be replaced with these selected items. A safety margin is defined for the minimum support, and the user is protected from false values being learned.

Another approach using unknowns is proposed in (Wang & Jafari, 2005) with two modification strategies. Although the database scan in this approach is limited, it has a drawback. Only rules containing sensitive items on LHS can be hidden.

Some other studies extend heuristics and use border theory (Mannila & Toivonen, 1997). Since the itemsets on the border give a boundary between the frequent and the infrequent itemsets, these approaches do not take all itemsets into account and focus on maintaining the non-sensitive border itemsets.

The work in (Sun & Yu, 2007) is the first one introducing border based approach for itemset hiding. During the hiding process, a weight is assigned to elements of the expected positive border for being affected by item deletion. For the candidate item, the sum of weights of positive border itemsets is calculated, and the candidate item with minimal impact on the positive border is selected for deletion.

In (Moustakides & Verykios, 2008), heuristics are proposed using revised positive and negative borders while hiding itemsets. The idea is that, we can minimize the impact of the changes in the data by considering only to minimize the impact on the positive border of the frequent patterns. Maximizing the minimum gain, all the non-sensitive frequent itemsets which are not in the positive border remain above the support threshold, which means that they are preserved. For each item of sensitive itemset, a list of positive border itemsets depending on it is created. The itemset with the maximum distance from the border is selected, which is called the max-min

itemset. The algorithms try to modify the item affecting the support of the max-min itemset minimally.

In (Quoc Le et al., 2013), the authors proposed a heuristic approach based on intersection lattice theory and distance concepts for hiding sensitive association rules. The optimal distance from the top of the intersection lattice of frequent itemsets to the sensitive association rules and to the non-sensitive association rules is computed for hiding rules with the least side effect.

Heuristic approaches are fast but may have side effects, and the number of non-sensitive itemsets accidentally hidden may increase. To cope with this, exact approaches deal with the problem as a Constraint Satisfaction Problem (CSP). These approaches present better solutions in terms of the number of lost itemsets but have more complexity and may have a longer runtime.

The first itemset hiding approach based on constraint programming is in (Menon et al., 2005). In this approach, first, constraints for integer programming are defined. Solving the problem would lead us to identify the selection of transactions to be modified. Following this, heuristically, items are selected from the transactions and altered. This process continues until the selected transaction no longer supports any sensitive itemsets.

In (Gkoulalas-Divanis & Verykios, 2006), the authors defined distance measures for the sanitized database. Instead of the number of transactions, they considered the number of modified items. Minimization of this distance is accomplished by maximizing the occurrences of items of sensitive itemsets. Using the positive and negative borders and the Apriori property, constraints are defined to maximize itemset occurrences and minimize item modifications. The authors also propose an approach for the degree reduction of constraints. When the constructed CSP is not solvable, this approach removes one constraint and constructs the CSP again iteratively until the CSP is solvable.

In (Gkoulalas-Divanis & Verykios, 2009), the authors revised the previous approach and gave a two-phase iterative approach. Firstly, sensitive itemsets are hidden using the revised positive border of itemsets. Secondly, transactions are modified to support accidentally hidden itemsets. For both phases, CSP is used.

In (Ayav & Ergenc, 2015), the authors defined new constraints and relaxation procedures to provide an exact solution. This approach observes all frequent itemsets to ensure they are kept frequent after sanitization. Therefore, constraints for all frequent itemsets are created, but it is not efficient to apply on large datasets. The proposed approach ensures that the constraint solver is executed once. Instead of reconstructing constraints for unsolvable CSPs, relaxation variables are used.

There are also some techniques for itemset hiding based on evolutionary algorithms in recent years. Since the solution is NP-hard, dealing with the problem as an optimization problem is feasible. In (C.-W. Lin et al., 2014), an algorithm is proposed to hide sensitive itemsets through transaction deletion. Three side effects are used as weights. Hiding failures, missing itemsets, and artificial itemsets are evaluated to determine the transactions to be deleted for hiding sensitive itemsets. In (Lin et al., 2016), authors proposed particle swarm optimization-based algorithms, which need fewer parameters to be set compared to previous algorithms. In (Khuda Bux et al., 2018), an algorithm was proposed to formulate an objective function that estimates the effect on non-sensitive rules with recursive computation. The main disadvantage of these approaches is that sanitization is done by deleting transactions.

The authors in (Lefkir et al., 2022) hide sensitive frequent itemsets by deleting single items in transactions rather than removing entire transactions. They perform two-level optimization for the minimization of side effects. At first, optimization is done at the transaction level to find a set of candidate transactions. Then, the evaluation level determines items that should be removed from each transaction.

A summary of the literature for itemset hiding approaches mentioned in this chapter is given in Table 2.1 including the proposed approach in this thesis study.

Table 2.1 Summary of itemset hiding approaches

Algorithm	Itemset Hiding	Rule Hiding	Item Distortion	Item Blocking	Transaction addition/deletion	Heuristic	CSP based	GA based	Border	Lattice
(Atallah et al., 1999)	x		x			x				
(Dasseni et al., 2001)	x	x	x			x				
(Oliveira & Zaïane, 2002)	x		x			x				
(Guanling Lee et al., 2004)	x		x			x				
(Verykios et al., 2004)	x	x	x			x				
(Saygin et al., 2001)	x	x		x		x				
(Wang & Jafari, 2005)		x		x		x				
(Sun & Yu, 2007)	x		x						x	
(Moustakides & Verykios, 2008)	x		x						x	
(Quoc Le et al., 2013)		x	x							x
(Menon et al., 2005)	x		x				x			
(Gkoulalas-Divanis & Verykios, 2006)	x		x				x		x	
(Gkoulalas-Divanis & Verykios, 2009)	x		x		x		x		x	
(Ayav & Ergenc, 2015)	x		x				x			
(C.-W. Lin et al., 2014)	x				x			x		
(J. C.-W. Lin et al., 2016)	x				x			x		
(Khuda Bux et al., 2018)		x			x			x		
(Lefkir et al., 2022)	x		x					x		
(Yildiz et al., 2022)	x		x				x			

CHAPTER 3

BACKGROUND INFORMATION

3.1 Overview of Data Mining

The simplest definition of data mining is extracting knowledge from large amounts of data. Data mining is a natural result of evolution in technology. Data collection is increasing with available devices and cheaper storage options. The processing capabilities of computers are also increasing. Using data mining tools, we can turn this data into valuable information or knowledge. The simple goal of data mining is to predict or to learn, thus data mining tasks are categorized into two predictive tasks and descriptive tasks. Descriptive data mining tasks describe data with its general properties. Predictive data mining tasks make inferences based on known results found in data. Some of data mining techniques for accomplishing such tasks are given in following titles.

3.1.1 Classification

Classification techniques aim to predict labels of new data based on previous data, serving predictive tasks. Classification maps data into predefined groups or classes (Dunham, 2003). Since the classes are determined before the data is analyzed, classification is referred to as supervised learning. Based on the training set, the properties of classes are defined and used for classifying new data (Han & Kamber, 2006).

In classification by decision tree induction, a tree model is constructed from training data such that internal nodes represent test and leaf nodes hold labels or classes. Popular algorithms are ID3 (J. R. Quinlan, 1986), C4.5 (J. R. Quinlan, 1993), CART (Breiman, 1984). K-Nearest-Neighbor classifier (Altman, 1992) compares training tuples and unknown tuples searching closest ones in terms of a distance metric like Euclidian distance. Bayes classifiers use the Bayes theorem and predict membership

probabilities to a class. Naïve Bayes classifier (Domingos & Pazzani, n.d.) is one of the most popular ones being useful and having performance comparable with other classifiers. Neural network based classifiers use artificial neural networks; biologically inspired computational methods, for prediction (Hopfield & Tank, 1985). These classifiers have higher computational cost than the ones mentioned above.

3.1.2 Clustering

The process of grouping the data into classes or clusters is clustering. Clustering techniques aim to define groups in data serving descriptive tasks.The collection of the data object within a cluster has high similarity but is dissimilar to ones in other clusters (Han & Kamber, 2012). The groups are not predefined as it is in classification.

Partitioning methods for clustering construct k partitions and iteratively relocate data points for better clusters. K-means (Lloyd, 1982) and k-means++ (Arthur & Vassilvitskii, 2007) are popular algorithms. Hierarchical methods create a hierarchical decomposition of data objects. Algorithms of hierarchical cluster analysis are divided into two categories: divisible algorithms and agglomerative algorithms. The agglomerative approach starts with objects to form groups, and the divisive approach starts with one group and divides to form clusters. Density-based approach in clustering assumes that clusters are regarded as dense regions of objects in the data space that are separated by regions of low object density. These approaches come over the problem of methods that form spherical shapes and miss the discovery of arbitrarily shaped clusters. DBSCAN (Ester et al., 1996) is one of the popular density-based approaches.

3.1.3 Association Rule Mining

Frequent pattern mining techniques find patterns and associations in the dataset. Association rule mining discovers relations and patterns among items in datasets.

Firstly, frequent itemsets are found following that rules are generated from these itemsets. Formal definition is as follows:

Given a set of items $I=\{I_1, I_2, \ldots, I_m\}$ and a database of transactions $D=\{t_1,t_2,\ldots,t_n\}$ where $t_i=\{I_{i1}, I_{i2}, \ldots,I_{ik}\}$ and $I_{ij} \in I$ and X,Y are set of items, the association rule problem is to identify all association rules $X \rightarrow Y$ with a minimum support and confidence where support of association rule $X \rightarrow Y$ is the percentage of transactions in the database that contain $X \cup Y$ and confidence is the ratio of support of $X \cup Y$ to support of X.

The first part is more time-consuming, and the main concentration is to make it faster. In addition, finding frequent itemsets is solely valuable and usable. The roots can be followed to market basket analysis which became possible by the barcode technology. Therefore, frequent itemset/pattern mining and market basket analysis are used as synonyms.

It was first mentioned in (Agrawal et al., 1993). There are many approaches, but they are derivatives of the following popular ones: Apriori (Agrawal & Srikant, 1994), ECLAT (Zaki et al., 1997), and FP-Growth(Han et al., 2004). Most of the algorithms proposed are deviations from these three.

3.2 Overview of Privacy Preserving Data Mining

As data processing and collection capabilities increase, individuals' sensitive private information is seen under threat of data mining. In contrast, data mining is a very efficient tool for knowledge discovery. The raw data that is input for data mining applications or the results of these applications may contain sensitive information. Continuing data mining without violating the privacy of the owner's private data or sensitive knowledge requires some techniques that are titled under the name Privacy Preserving Data Mining. Input privacy protection and output privacy protection are two main branches. In Figure 3.1 hierarchy of PPDM techniques is given.

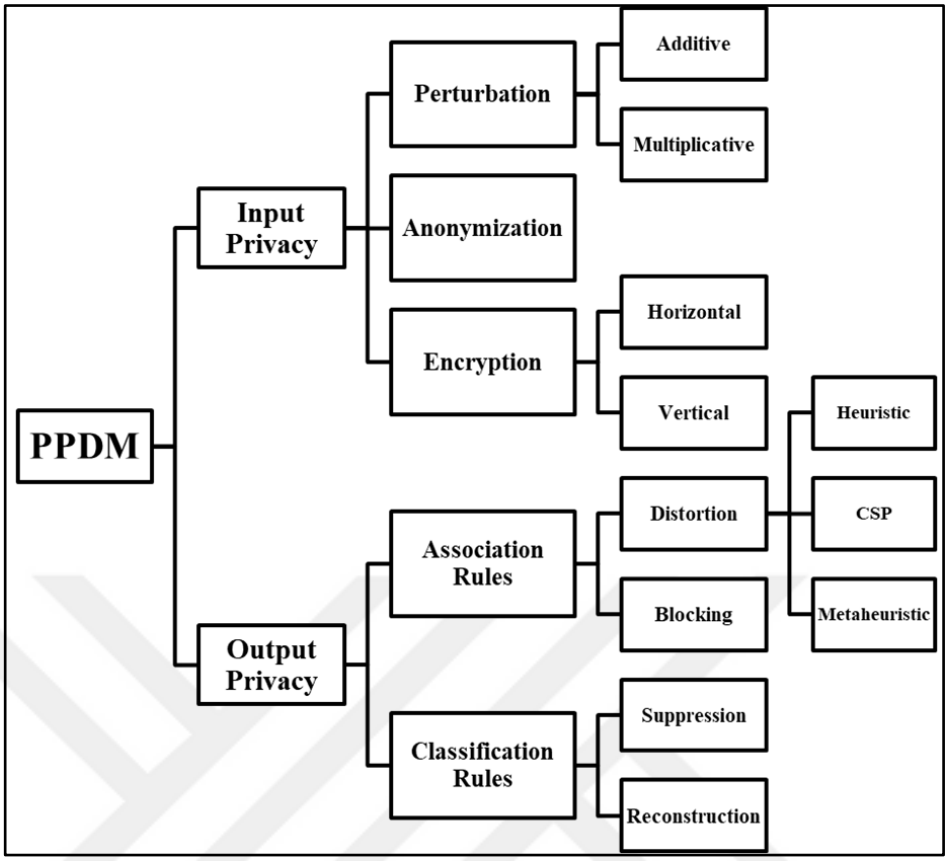


Figure 3.1 PPDM hierarchy

3.2.1 Input Privacy Protection

Input privacy protection in data mining includes techniques like perturbation, anonymization, and encryption. These techniques aim protecting private data in such a way that sanitized database is still valuable for data mining.

Perturbation

These approaches are based on distorting the raw data, then providing this distorted dataset as input to the data mining algorithm. Therefore, one can not easily reveal individually identifiable values. Additive perturbation techniques add randomized noise such that the overall distribution of data can be discovered while individual points can not be identified. Multiplicative perturbation distorts values by random projection or random rotation techniques.

Anonymization

These approaches aim to prevent individual sensitive information can not be identified with the help of other identifiers. Anonymized data has less granularity. In k-anonymity, identifier attributes are modified in such a way that they turn out to be indistinguishable for k records, where k > 1. L-diversity is an addition to k-anonymity. It aims sensitive identifier attributes to have diverse values.

Encryption

These approaches use secure and cryptographic protocols for the distribution of information between different parties. Techniques differ in horizontally partitioned data and vertically partitioned data.

3.2.2 Output Privacy Protection

Input privacy protection in data mining includes techniques like association rule hiding and classification rule hiding. These techniques aim protecting sensitive data mining results being revealed from sanitized database and it is still valuable for data mining.

Association Rule Hiding

Associations rules or frequent patterns obtained from association rule mining algorithms may contain sensitive knowledge. Some modification on the dataset is needed for privacy protection without effecting non-sensitive rules or patterns. Literature review of approaches is given in previous chapter.

Association rule hiding approaches are based on distortion or blocking of items in the database. Distortion based association rule hiding algorithms aim to reduce support or confidence of the sensitive rules. The transactions are modified by deleting some items or adding new items. In Figure 3.2 an example is given.

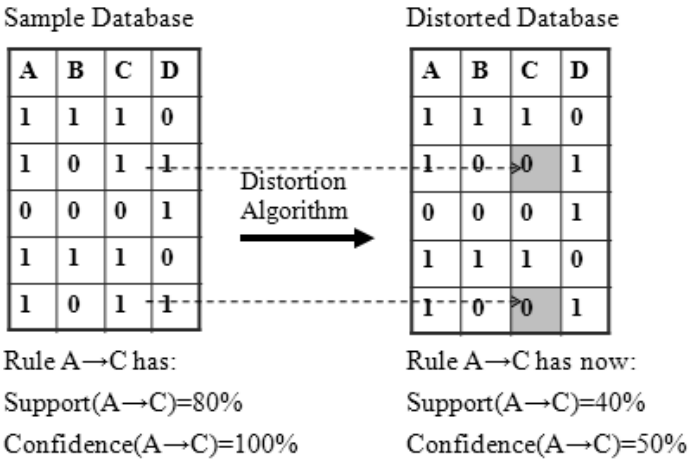


Figure 3.2 Distortion based rule hiding example

Blocking based algorithms use unknowns to keep support and confidence of rules in an interval. Instead of adding or deleting items, unknown values are added to sanitized database. An example is given in Figure 3.3.

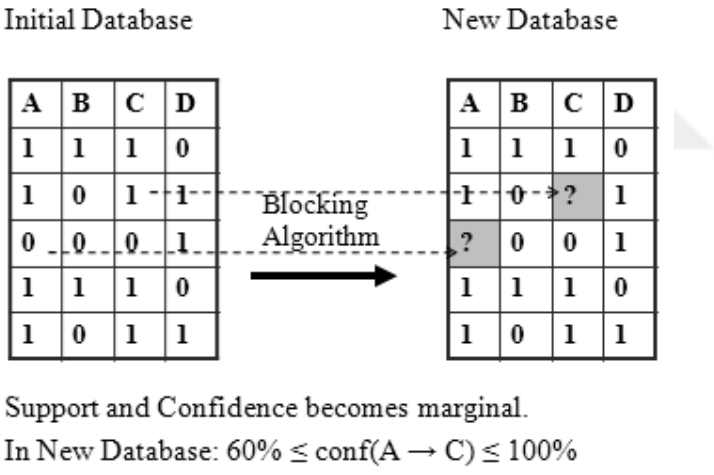


Figure 3.3 Blocking based rule hiding example

Classification rule hiding

Classification rule hiding algorithms consider a set of classification rules as sensitive, like association rule mining algorithms. Classification rule hiding methods have two main branches. These are suppression-based techniques and reconstruction-based techniques.

- Suppression-Based Techniques aim at reducing the confidence of classification rules which are defined as sensitive. The sanitization is done by distorting the values of some attributes in the original database that are related to the existence of sensitive rules. (Chang & Moskowitz, 1998) address the problem of inference caused by the downgrading of data in classification decision rules. A blocking technique called parsimonious downgrading is applied. By this technique, the inference channels those breach privacy of sensitive classification rules are blocked. This blocking process consists of modifying transactions such that some missing values appear in the database that is published.
- Reconstruction-Based Techniques target reconstructing the original database by using only supporting transactions of non-sensitive rules. An approach for reconstructing the database is (Natwichai & Orlowska, 2006). The algorithm works as follows. Firstly a set of valid classification rules is generated from the original database and presented to the data publisher for sensitivity check. Then, a decision tree classifier that contains only non-sensitive rules is constructed and a database confirming this tree is reconstructed for publishing. The newly reconstructed database is similar to the original one, except from the sensitive part. This database holds the non-sensitive rules but does not hold the sensitive ones. Thus, it is safe to publish this database.

CHAPTER 4

ITEMSET HIDING USING SIBLING ITEMSET CONSTRAINTS

4.1 Preliminaries

The basic concepts can be defined as follows. Let $I = \{i_{1,}, i_2, \dots i_m\}$ be a set of literals, called items. Let $D = \{T_{1,}, T_2, \dots T_n\}$ be a database of transactions where each transaction T_i is a set of items in Isuch that $T_i \subseteq I$. Each transaction can be defined in the binary form where $d_{ij} = 1$ if the j-th item of I appears in the transactiont_i. Considering all transactions, for ease of calculation, we have a binary form of D as a matrix that is called bitmap notation. It is given in Equation 4.1.

$$d_{ij} = \begin{cases} 1, & if\ I_j \in T_i \\ 0, & otherwise \end{cases} \tag{4.1}$$

Let X be a set of items where $X \subseteq I$. We call it an itemset. If $X \subseteq t_i$ then itemset X is said to be supported by transaction t_i. In other words, all items of the itemset appear in the transaction. The number of transactions in D supporting itemset X is defined as the support count of X. Support count of itemset Xin bitmap notation can be calculated as given in Equation 4.2.

$$\sigma(X) = \sum_{i=1}^{n} \prod_{I_j \in X} d_{ij} \tag{4.2}$$

If the support count of itemset X is at least equal to the minimum support count; $\sigma(X) \geq \sigma_{min}$, then itemset X is called frequent or large. The frequent itemset mining problem is to find all frequent itemsets in the database for a predefined minimum support threshold. We can define the set of all frequent itemsets F, as stated in Equation 4.3.

$$F = \{X \subseteq I : \sigma(X) \geq \sigma_{min}\} \tag{4.3}$$

Some itemsets in F may contain sensitive information. Denoting these as$S$ referring to sensitive itemsets, we need to adjust database DintoD^s in such a way that frequent itemsets of sanitized database F^sexcludes sensitive itemsets. As known from Apriori property, if an itemset is frequent, all of its subsets are also frequent. Rephrasing vice versa for the itemset hiding concept, we can say that when an itemset is sensitive, its supersets are also sensitive. Sensitive supersets Ss should also be hidden, which can be defined in Equation 4.4.

$$Ss = \{X \in F : \forall Y \in S, \quad X \supset Y\} \tag{4.4}$$

The remaining frequent itemsets are non-sensitive frequent itemsets donated by F^n is given in Equation 4.5.

$$F^n = F - (S \cup Ss) \tag{4.5}$$

Then we can define frequent itemset hiding problem as modifying database D into D^s in such a way that F^s -frequent itemsets of sanitized database D^s - excludes sensitive frequent itemsets S whereas non-sensitive frequent itemsets F^n can still be mined from D^s with the same minimum support threshold.

$$F^s = \{X \subseteq I : X \in F^n \; and \; \sigma^s(X) \geq \sigma_{min}\} \tag{4.6}$$

For an ideal sensitive itemset hiding methodology, as many as the following goals should be accomplished on the sanitized database with the same minimum support threshold.

1. Modification of the database is minimized. Such that originality of the database is kept as much as possible.
2. All sensitive itemsets are hidden and don't appear in the sanitized database.

3. Supersets of sensitive itemsets are also hidden and don't appear in the sanitized database. We know from the Apriori property that this goal is also accomplished if the 1st goal is achieved.
4. All non-sensitive frequent itemsets appear in the sanitized database. If an itemset doesn't appear in the new database, it is called a lost itemset.
5. No new itemset appears in the sanitized database. Such itemsets are called ghost itemsets; however, approaches that delete items from the dataset naturally accomplish this goal, and no new itemsets can be mined.

Goal 1 can be rewritten as accomplishing $\min(D - D^s)$. Minimization of modification for approaches use item deletion; we can say that number of items deleted should be minimized. Using the bitmap notation given in Equation 4.1, let us define items in the new dataset as d_{ij}^s. Then, the minimization of the number of 1s converted to 0 is the maximization of the 1s in D^s and can be defined as follows.

$$max \sum_{i,j} d_{ij}^s \tag{4.7}$$

Goal 2 can be accomplished by keeping the support count of all sensitive itemsets below the minimum support count in the new dataset.

$$\begin{gathered} \forall X \in S \\ \sigma^s(X) < \sigma_{min} \end{gathered} \tag{4.8}$$

Goal 3 is accomplished if goal 2 is already satisfied.

Goal 4 can be accomplished by keeping the support count of all non-sensitive itemsets at the same or above the minimum support count in the new dataset.

$$\begin{gathered} \forall X \in F^n \\ \sigma^s(X) \geq \sigma_{min} \end{gathered} \tag{4.9}$$

Goal 5 is satisfied if the approach uses item deletion for the sanitization method and doesn't add any item to the new dataset. Some approaches use reconstruction methods and may also add new transactions to the sanitized dataset. Such approaches may be exposed to this side effect.

$$\begin{gathered} \forall X \in F^s \\ X \in F^n \end{gathered} \tag{4.10}$$

The majority of sensitive itemset hiding approaches aim to hide sensitive itemsets while minimizing modified items in the dataset and the number of lost itemsets. They are focused on goals 1, 2, 3, and 4.

4.2 CSP Formulation

Preliminaries for sensitive itemset hiding are already given, and this process can be formulated as a constraint satisfaction problem. In satisfying goal 1, we can say that there are two kinds of constraints. The first type of constraint defined for accomplishing goal 2 is defined in Inequation 4.8. Other constraints are determined to achieve goal 4, given in Inequation 4.9. The first type is compulsory since hiding sensitive itemsets is the primary goal of frequent itemset hiding. The second type serves for the preservation of the non-sensitive frequent itemsets.

For CSP formulation, we modify the dataset into an intermediate form. Consider X as one of the sensitive itemsets. Then all transactions supporting X should be modified to an intermediate state for constraint formulation. The items of the sensitive itemset are modified to temporary binary u variables. Using the bitmap notation, this modification is given in the intermediate dataset and can be defined as shown in Equation 4.11.

$$\begin{gathered} \forall X \in S \\ \sim d_{ij} = \begin{cases} u_{ij}, & if\ X \subseteq Ti\ and\ I_j \in X \\ d_{ij}, & otherwise \end{cases} \end{gathered} \tag{4.11}$$

Since the items that may be modified in the sanitized dataset are u variables, the optimal itemset hiding problem can be formulated as follows.

$$maximize\left(\sum_{u_{ij}\in U} u_{ij}\right)$$
$$subject\ to \begin{cases} \forall X \in S, \sigma^s(X) < \sigma_{min} \\ \forall Y \in F^n, \sigma^s(Y) \geq \sigma_{min} \end{cases} \tag{4.12}$$

4.3 Finding Sibling Itemset Constraints

Frequent itemset mining and CSP formulation preliminaries are given in the previous section. Considering all non-sensitive frequent itemsets will increase the number of constraints. To lessen constraints, we introduce the sibling itemset concept. Sibling itemsets SI of a frequent k-itemset X are generating itemsets of $k+1$ candidate itemset. The idea behind this concept is that hiding a k-itemset will also hide its $k+1$ supersets but remain non-sensitive subsets of these $k+1$ supersets discoverable. This represents a local border.

$$\forall X \in S,$$
$$SI(X) = \begin{cases} Y \in F^n : |Y - X| = 1 \\ and \\ Y \equiv X \end{cases} \tag{4.13}$$

Using sibling itemsets instead of all non-sensitive frequent itemsets, CSP defined in (4.12) can be defined as follows

$$maximize\left(\sum_{u_{ij}\in U} u_{ij}\right)$$
$$subject\ to \begin{cases} \forall X \in S, \sigma^s(X) < \sigma_{min} \\ \forall Y \in SI, \sigma^s(Y) \geq \sigma_{min} \end{cases} \tag{4.14}$$

Generation and determining support of sibling itemsets of a sensitive itemset is conducted in the hiding process. In this way, the time consumption of prior itemset mining is eliminated.

There are two types of constraints for our CSP: sensitive itemset constraints and sibling itemset constraints. The first type ensures that sensitive itemsets are below the defined minimum support threshold. Thus, all of these constraints must be satisfied. The second type of constraint is satisfied to lessen information loss. There are situations when all of these can not be satisfied, and CSP is not solvable. Then, we need to sacrifice some of them. In our approach, information loss is preferred to a privacy breach. Therefore, some constraints for sibling itemsets can be sacrificed. Instead of removing any of those constraints, we add binary relaxation variables. By doing this, we do not need to reformulate CSP and run the solver more than once. We add a unique binary relaxation variable r to the inequality for all sibling constraints.

$$maximize\left(\sum_{u_{ij}\in U} u_{ij}\right) \qquad (4.15)$$
$$subject\ to\begin{cases} \forall X \in S, \sigma^s(X) < \sigma_{min} \\ \forall Y \in SI, \sigma^s(Y) + r_Y \geq \sigma_{min} \end{cases}$$

Relaxation on constraints should be minimized to ensure that information loss is minimized.

$$minimize\left(\sum_{Y\in SI} r_Y\right) \qquad (4.16)$$

So Equation 4.17 gives our final CSP formulation.

$$maximize\left(\sum_{u_{ij}\in U} u_{ij} - \sum_{Y\in SI} r_Y\right)$$
$$subject\ to \begin{cases} \forall X \in S, \sigma^s(X) < \sigma_{min} \\ \forall Y \in SI, \sigma^s(Y) + r_Y \geq \sigma_{min} \end{cases} \quad (4.17)$$

4.4 Illustrative Example

In the following, an illustrative example of our hiding approach is given. Let D be the dataset of 10 transactions shown in Table 4.1. Our set of items is $I = \{A, B, C, D, E\}$

Table 4.1 Dataset D

TID	Items
T_1	AC
T_2	ACDE
T_3	CD
T_4	BE
T_5	ACDE
T_6	DE
T_7	C
T_8	AB
T_9	AC
T_{10}	CD

Using the bitmap notation given in (4.1), we have a 10x5 binary matrix representation of D.

Table 4.2 Dataset D in bitmap notation

A	**B**	**C**	**D**	**E**
1	0	1	0	0
1	0	1	1	1
0	0	1	1	0
0	1	0	0	1
1	0	1	1	1
0	0	0	1	1
0	0	1	0	0
1	1	0	0	0
1	0	1	0	0
0	0	1	1	0

Using the formulation in (4.2), we can calculate the support count of an itemset. For instance, support count of itemset $\{AC\}$

$$\sigma_{\{AC\}} = d_{1,1}d_{1,3} + d_{2,1}d_{2,3} + \cdots + d_{10,1}d_{10,3}$$
$$\sigma_{\{AC\}} = 4$$

Given minimum support count $\sigma_{\min} = 2$ and Equation (4.3), we can find 16 frequent itemsets.

$$F = \{A, B, C, D, E, AC, AD, AE, CD, CE, DE, ACD, ACE, ADE, CDE, ACDE\}$$

Suppose that itemset $\{CD\}$ is given as sensitive and needs to be hidden. Then $S = \{CD\}$

Equations (4.4) and (4.5) give supersets of sensitive itemsets and non-sensitive itemsets as follows:

$$Ss = \{ACD, CDE, ACDE\}$$
$$F^n = \{A, B, C, D, E, AC, AD, AE, CE, DE, ACE, ADE\}$$

All itemsets in S and Ss must be hidden to achieve privacy, whereas as many itemsets as in F^nshould remain frequent after sanitization.

Using the formulation given in (4.11), transactions supporting itemset $\{CD\}$ are modified with binary variables. Their values will be determined after CSP is solved. The intermediate form of the dataset is given in Table 4.3.

Table 4.3 Intermediate form of dataset D

A	B	C	D	E
1	0	1	0	0
1	0	$\mathbf{u_{2,3}}$	$\mathbf{u_{2,4}}$	1
0	0	$\mathbf{u_{3,3}}$	$\mathbf{u_{3,4}}$	0
0	1	0	0	1
1	0	$\mathbf{u_{5,3}}$	$\mathbf{u_{5,4}}$	1
0	0	0	1	1
0	0	1	0	0
1	1	0	0	0
1	0	1	0	0
0	0	$\mathbf{u_{10,3}}$	$\mathbf{u_{10,4}}$	0

Using (4.13), we can find sibling itemsets as $SI = \{AC, CE, AD, DE\}$. Now we can define the CSP formulation given in (4.17)

$$maximize u_{2,3} + u_{2,4} + u_{3,3} + u_{3,4} + u_{5,3} + u_{5,4} + u_{10,3} + u_{10,4} - r_{\{AC\}} - r_{\{CE\}} - r_{\{AD\}} - r_{\{DE\}}$$

$$subject to \begin{cases} u_{2,3}u_{2,4} + u_{3,3}u_{3,4} + u_{5,3}u_{5,4} + u_{10,3}u_{10,4} < \sigma_{min} \\ 1 + u_{2,3} + u_{5,3} + r_{\{AC\}} \geq \sigma_{min} \\ u_{2,3} + u_{5,3} + r_{\{CE\}} \geq \sigma_{min} \\ u_{2,4} + u_{5,4} + r_{\{AD\}} \geq \sigma_{min} \\ u_{2,4} + u_{5,4} + r_{\{DE\}} \geq \sigma_{min} \end{cases}$$

$$where\ \sigma_{min} = 2$$

The solution of such CSP is

$$u_{2,3} = u_{2,4} = u_{3,4} = u_{5,3} = u_{10,4} = r_{\{AD\}} = 1$$
$$u_{3,3} = u_{5,4} = u_{10,3} = r_{\{AC\}} = r_{\{CE\}} = r_{\{DE\}} = 0$$

When results are applied to the intermediate form of the dataset, we obtain the sanitized datasetD^sas given in Table 4. From this sanitized dataset, we can find itemsets for minimum support count $\sigma_{\min} = 2$ as $F^s = \{A, B, C, D, E, AC, AE, CE, DE, ACE\}$.

The sensitive itemset $\{CD\}$ is no longer frequent for support count 2. Compared to the initial dataset number of itemsets decreased to 10 from 16. 2itemsets are accidentally lost, and 3 itemsets are supersets of $\{CD\}$; therefore, they are also missing.

Table 4.4 Sanitized Dataset

A	B	C	D	E
1	0	1	0	0
1	0	1	1	1
0	0	**0**	1	0
0	1	0	0	1
1	0	1	**0**	1
0	0	0	1	1
0	0	1	0	0
1	1	0	0	0
1	0	1	0	0
0	0	**0**	1	0

4.5 Experimental Analysis

In this section, we give a performance evaluation of our approach. The reference algorithm for comparison is IPA (Gkoulalas-Divanis & Verykios, 2006). We implemented the algorithms using Python. Constraints are solved using Minizinc (Nethercote et al, 2007). Implementations use Pymzn (Dragone, 2022) library to be able to invoke, run and gather results from the constraint solver. All computational experiments are conducted on a PC running MS Windows 10 with an Intel i5-4200U CPU and 8 GB RAM.

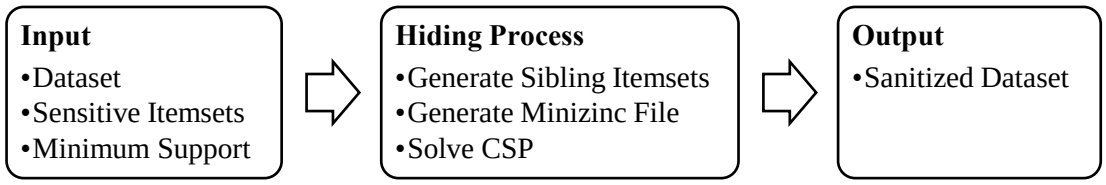


Figure 4.1. Itemset Hiding Framework Using Sibling Itemsets

The whole process of sanitization of a dataset by proposed approach in this thesis study is given in Figure 4.1. Main differences between proposed approach and IPA algorithm are: IPA needs frequent itemsets as input for border construction, IPA generates revised border for constraints in hiding process.

4.5.1 Itemset Hiding Evaluation Metrics

Itemset hiding aims to transform the dataset in a way that sensitive itemsets are concealed, non-sensitive frequent itemsets are preserved, ghost itemsets are not generated, and dataset distortion is minimum. These goals can be measured respectively as given in the following subtitles.

Hiding Failure

This metric concerns sensitive itemsets remaining frequent after the sanitization process. It is defined as the percentage of sensitive itemsets that appear in the sanitized dataset divided by the ones that appeared in the original dataset.

$$HF = \frac{|S \cap F^s|}{|S|} \tag{4.18}$$

The proposed approach ensures that all sensitive itemsets are hidden; therefore, $HF = 0$ for all scenarios. As far as we have surveyed, all proposed approaches focus on *HF* and ensure that it is 0. Our reference algorithm IPA also ensures that all sensitive itemsets are hidden and has no hiding failure.

Artifactual Patterns

This metric also concerns the side effects of the sanitization process because some approaches insert items or transactions to the dataset during or after sanitization. Thus, ghost itemsets may be generated in sanitized dataset. It is calculated as the ratio of itemsets that did not appear in the original dataset but appeared in the sanitized dataset to the itemsets that appear in both the original and the sanitized datasets. In other words ghost itemset generation.

$$AP = \frac{|F^s \cup F|}{|F|} \tag{4.19}$$

Only the approaches inserting new items into the dataset are prone to this side effect. Since the proposed approach does not insert items on the original dataset, it is not possible to produce new itemsets from the sanitized dataset. This is the same for the IPA algorithm.

Dissimilarity

This metric is the measure of the differences between the original and the sanitized dataset quantified by comparing the number of items added or deleted.

$$Diss(D, D^s) = \sum_{i=1}^{n} \sum_{j=1}^{m} \begin{cases} 0 \; if \; d_{ij} = d_{ij}^s \\ 1 \; if \; d_{ij} \neq d_{ij}^s \end{cases} \tag{4.20}$$

The proposed approach, the number of deleted items gives dissimilarity between the original and sanitized dataset. The number of deleted items are identical with IPA algorithm. Some heuristic approaches may prone to this side effect since they assume that sensitive itemsets are mutually exclusive and have no intersection.

Misses Cost

This metric concerns the side effects of the sanitization process. It is measured as the percentage of non-sensitive patterns that disappeared in the sanitized dataset divided by the ones that appeared in the original dataset. It gives the ratio of preservation level of non-sensitive itemsets.

$$MC = \frac{|F^n| \; - |F^s|}{|F^n|} \tag{4.21}$$

We have given this measure as the number of lost itemsets. This is the only metric that differs with the IPA algorithm and further comparison is given in 4.5.2.

In (Bertino et al., 2005) authors give an evaluation framework for compassion of PPDM algorithms. They identified five important evaluation dimensions:

- Efficiency: The ability of algorithm to execute with good performance with available sources.
- Scalability: The ability of algorithm handling increasing size of data to be sanitized.
- Data quality: Preservation of original data values and of data mining results after the application of a privacy preserving technique
- Hiding failure: The portion of sensitive information that is failed to be hidden by the application of a privacy preservation technique;
- Privacy level: The degree of inferring hidden knowledge that has been hidden, can still be predicted.

Last three have been mentioned so far with itemset hiding perspective. Not all studies give evaluations for efficiency and scalability. Although computation capabilities have been increased last decades an approach can be only usable if it can be executed with available sources. Therefore, this study also focuses on these aspects. In 4.5.2 regarding to these aspects, runtime comparison with a reference approach is also given. Having different characteristics of datasets and different hiding scenarios results and discussion is given.

4.5.2 Comparison with a Reference Approach

We evaluated the algorithms on six different datasets obtained from (Goethals, 2022). Characteristics of these datasets are given in Table 4.5. Since the IPA algorithm uses frequent itemsets discovered before the hiding process, we also provide time consumption for tested values on datasets. Python implementation (Borgelt, 2022) of the Eclat (Borgelt, 2003) algorithm is used for frequent itemset mining.

Table 4.5 Properties of datasets

Dataset Name	Number of Transactions	Average Transaction Length	Number of Items	Minimum support count	Number of Frequent itemsets	Runtime (seconds)
T10I4D100K	100000	10.10	870	500(%0.5)	1073	9.15
T40I10D100K	100000	39.60	942	500(%0.5)	1286037	392.96
Mushroom	8124	23.00	119	406(%5)	3755704	9.77
retail	88162	10.30	16470	440(%0.5)	581	2.00
BMS1	59602	2.51	497	60(%0.1)	3991	0.88
BMS2	77512	4.62	3340	77(%0.1)	24143	5.22

We experiment with the algorithms using different hiding scenarios: hiding 1 2-itemset (HS_2.1), hiding 2 2-itemset(HS_2.2), hiding 3 2-itemset(HS_2.3), hiding 1 3-itemset(HS_3.1), hiding 2 3-itemset(HS_3.2), hiding 1 4-itemset(HS_4.1). The sensitive itemsets chosen have support counts close to the minimum support count since those itemsets are more logical to be hidden and indistinguishable compared to the rest. For ease of use in tables, our approach is named HISB(Hiding Itemsets using SiBlings) during this section.

The results of the evaluation for the T10I4100K dataset are given in Table 4.6. Columns represent hiding scenarios, side effects such as the number of lost itemsets, and running time in seconds for algorithm IPA and HISB. Both algorithms perform well in terms of several lost itemsets. In defined scenarios, no itemset is lost. Our approach performs better in terms of runtime in 4 scenarios. It should be noted that IPA needs prior itemset mining, which costs additional 9.15 seconds.

Table 4.6 Results of the T10I4100K dataset

Hiding Scenario	Number of Lost Itemsets(IPA/HISB)	Algorithm IPA (seconds)	Algorithm HISB (seconds)	HISB Runtime Advantage(%)
HS_2.1	0/0	5.72	4.04	42%
HS_2.2	0/0	6.75	5.2	30%
HS_2.3	0/0	7.62	6.03	26%
HS_3.1	0/0	6.78	5.54	22%
HS_3.2	0/0	9.51	10.31	-8%
HS_4.1	0/0	9.01	10.5	-14%

The results of the evaluation for the T40I10100K dataset are given in Table 4.7. IPA performs better in terms of several lost itemsets. On the other hand, our approach

performs better in terms of runtime even though prior itemset mining consumption is not included for IPA.

Table 4.7 Results of the T40I10100K dataset

Hiding Scenario	Number of Lost Itemsets(IPA/HISB)	Algorithm IPA (seconds)	Algorithm HISB (seconds)	HISB Runtime Advantage(%)
HS_2.1	0/1	13.64	13.31	2%
HS_2.2	0/1	27.59	14.59	89%
HS_2.3	0/2	62.49	22.88	173%
HS_3.1	0/0	95.61	18.92	405%
HS_3.2	0/0	1110.97	31.32	3447%
HS_4.1	0/1	408.48	24.02	1601%

The results of the evaluation for the Mushroom dataset are given in Table 4.8. Both algorithms perform well regarding the number of lost itemsets where no itemset is lost. On the other hand, our approach performs better in terms of runtime even though prior itemset mining consumption is not included for IPA.

Table 4.8 Results of the mushroom dataset

Hiding Scenario	Number of Lost Itemsets(IPA/HISB)	Algorithm IPA (seconds)	Algorithm HISB (seconds)	HISB Runtime Advantage(%)
HS_2.1	0/0	8.98	1.56	476%
HS_2.2	0/0	18.45	2.6	610%
HS_2.3	0/0	22.45	4.23	431%
HS_3.1	0/0	23	2.67	761%
HS_3.2	0/0	25.78	4.96	420%
HS_4.1	0/0	17.44	6.11	185%

The results of the evaluation for the BMS1 dataset are given in Table 4.9. IPA performs better in terms of several lost itemsets. The runtime performance of hiding processes is close in 5 of 6 scenarios.

Table 4.9 Results of the BMS1 dataset

Hiding Scenario	Number of Lost Itemsets(IPA/HISB)	Algorithm IPA (seconds)	Algorithm HISB (seconds)	HISB Runtime Advantage(%)
HS_2.1	0/0	1.07	1.06	1%
HS_2.2	0/0	1.14	1.17	-3%
HS_2.3	0/1	1.18	1.18	0%
HS_3.1	0/0	1.2	1.36	-12%
HS_3.2	0/1	1.47	1.39	6%
HS_4.1	0/2	2.96	1.57	89%

The results of the evaluation for the BMS2 dataset are given in Table 4.10. Both algorithms perform well in terms of several lost itemsets. The runtime performance of hiding processes is similar in 4 scenarios.

Table 4.10 Results of the BMS2 dataset

Hiding Scenario	Number of Lost Itemsets(IPA/HISB)	Algorithm IPA (seconds)	Algorithm HISB (seconds)	HISB Runtime Advantage(%)
HS_2.1	0/0	5.53	5.49	1%
HS_2.2	0/0	5.61	5.57	1%
HS_2.3	0/0	5.82	5.75	1%
HS_3.1	0/0	15.25	5.93	157%
HS_3.2	0/0	18.9	6.53	189%
HS_4.1	0/0	6.04	6.01	0%

Results of the evaluation for the Retail dataset are given in Table 4.11. Both algorithms perform well in terms of several lost itemsets. Runtime performance of hiding processes is close.

Table 4.11 Results of the retail dataset

Hiding Scenario	Number of Lost Itemsets(IPA/HISB)	Algorithm IPA (seconds)	Algorithm HISB (seconds)	HISB Runtime Advantage(%)
HS_2.1	0/0	36.21	33.21	9%
HS_2.2	0/0	38.59	38.54	0%
HS_2.3	0/0	43.36	52.48	-17%
HS_3.1	0/0	39.15	34.89	12%
HS_3.2	0/0	43.92	42.98	2%
HS_4.1	0/0	45.11	44.52	1%

4.6 Discussion

First of all, we can say that using sibling itemsets constraints to lessen the runtime of the hiding process is effective. Even though comparison tables do not include itemset mining time consumption before the hiding process, our approach performs faster in most cases. To add this, in some cases, the number of border itemsets or the length of some border itemsets constructed by the IPA algorithm cause distinctive runtime differences in the hiding process. Experiments on the Mushroom dataset reveal that eliminating prior mining is advantageous when the dataset is dense. Although this dataset has fewer items and transactions, the number of frequent itemsets for the given support threshold is over 3.5 million. We also observed that unsolvable constraints cause another disadvantage. However, this is not common in most cases. Secondly, the number of lost itemsets caused by our approach is tolerable when considering the number of frequent itemsets.

At this juncture, we would like to mention that we have also implemented the algorithm given in (Ayav & Ergenc, 2015). It promises optimum results since it is a full exact approach and uses relaxation techniques for CSP. However, we could not finish the experiments because insufficient runtime or hardware limitations caused crashes. The reason for this problem is that the algorithm generates constraints for all non-sensitive frequent itemsets. Considering our experiments, it should generate constraints for over 1 million and 3 million frequent itemsets for T40I10D100K and Mushroom datasets, respectively, which is not feasible.

4.7 Development Milestones

Development of the proposed approach is given in this part. Firstly, constraint solver choice is discussed. It has been observed that different solvers may have different solutions or no solution in sufficient time. Following, evaluation on small datasets with a comparison to an exact approach is given, which is resulted in improvement on the proposed approach.

4.7.1 Constraint Solver

There are many solvers on the market, both commercial and free. We are using Minizinc. MiniZinc is a free and open-source constraint modeling language. It can be used to model constraint satisfaction and optimization problems in a high-level, solver-independent way. The model is then compiled into FlatZinc, a solver input language that is understood by a wide range of solvers. MiniZinc is developed at Monash University in collaboration with Data61 Decision Sciences and the University of Melbourne.

Minizinc bundle has many solvers included. Gecode, Chuffed, CBC, and G12MIP are four of them. Gecode and Chuffed solvers take more time than remaining two. Therefore, G12MIP and CBC are used in our evaluations.

G12MIP

NICTA's (National Information and Communications Technology Australia) G12 project aimed to develop a new constraint programming platform featuring a suite of languages. Zinc is a modeling language based on first-order logic with extensions to accommodate numerical constraints and commonly used data structures. It allows the model or generic domain description to be separated from the data or specifics of the problem instance. Mercury is the logic programming language of that name, used in G12 as a basis for constraint logic programming. It allows solvers of different kinds suitable for a particular problem, in our case, MIP solver.

CBC

Computational Infrastructure for Operations Research (COIN-OR) is a project that aims to "create for mathematical software what the open literature is for mathematical theory.". COIN-OR branch and cut (CBC or Cbc) is an open-source mixed integer programming solver written in C++. It can be used as both a stand-alone executable

and as a callable library (through A Mathematical Programming Language (AMPL), General Algebraic Modeling System (GAMS), MPL, AIMMS, or PuLP).

Evaluation of Different Solvers

We evaluated the algorithms on three different datasets from (Goethals, 2022). Characteristics of these datasets are given in Table 4.12 below.

Table 4.12 Dataset properties of constraint solver evaluation

Dataset Name	Number of Transactions	Average Transaction Length	Number of Items	Minimum support count	Number of Frequent itemsets
T10I4D100K	100000	10.10	870	500(%0.5)	1073
T40I10D100K	100000	39.60	942	500(%0.5)	1286037
Mushroom	8124	23.00	119	406(%5)	3755704

The results of the evaluations are given in Table 4.13. Columns representing dataset name, hiding scenario, and side effect as the number of lost itemsets for each two constraint solvers.

Table 4.13 Side effect of evaluation for constraint solvers

Dataset Name	Hiding Scenario	g12mip	cbc
T10I4D100K	HS_2.1	0	0
T10I4D100K	HS_2.2	0	0
T10I4D100K	HS_2.3	0	0
T10I4D100K	HS_3.1	0	0
T10I4D100K	HS_3.2	0	0
T10I4D100K	HS_4.1	0	0
T40I10D100K	HS_2.1	1	0
T40I10D100K	HS_2.2	2	0
T40I10D100K	HS_2.3	2	0
T40I10D100K	HS_3.1	0	41
T40I10D100K	HS_3.2	0	0
T40I10D100K	HS_4.1	1	2
Mushroom	HS_2.1	0	0
Mushroom	HS_2.2	0	436
Mushroom	HS_2.3	0	122
Mushroom	HS_3.1	1	1
Mushroom	HS_3.2	1	9
Mushroom	HS_4.1	1	1

We experiment the algorithms using different hiding scenarios: hiding 1 2-itemset (HS_2.1), hiding 2 2-itemset(HS_2.2), hiding 3 2-itemset(HS_2.3), hiding 1 3-itemset(HS_3.1), hiding 2 3-itemset(HS_3.2), hiding 1 4-itemset(HS_4.1).

In T10I4D100K dataset, both solvers are identical. None affected the number of lost itemsets. In T40I10D100K dataset, cbc solver performed close to g12mip in all scenarios except hiding 1 3-itemset(HS_3.1). In this scenario, g12mip does not produce lost itemsets; however, cbc produced 41 itemsets. In the mushroom dataset uneven performance of cbc solver becomes worse. In HS_2.2 and HS_2.3 scenarios, g12mip does not produce any lost itemsets but cbc produced 436 and 122 lost itemsets, respectively. Using g12mip constraint solver is more consistent than using cbc solver for our approach in these test cases.

4.7.2 Evaluation on Small Datasets

We have mentioned that the approach in (Ayav & Ergenc, 2015) is one of the most comprehensive since it generates constraints for all frequent itemsets. However, this is not feasible in scenarios and datasets we have worked with. We have stated this previously. Therefore, we conducted a new evaluation with small datasets and large minimum support thresholds. During the test, it is observed that some sensitive itemsets may not have any siblings, so no constraint is generated for them. This was not an issue in large datasets. However, we need to modify the approach and include *k-1* subsets of a sensitive *k*-itemset. Sibling generation formulation for such itemsets is given in Equation 4.22.

$$SI(X) = \{\} \Rightarrow SI(X) = \{Y \in F^n : |X - Y| = 1 \; and \; Y \subseteq X\} \qquad (4.22)$$

For ease of understanding, the approach in (Ayav & Ergenc, 2015) is named AE15 and modified HISB approach is named HISB+S (Hiding Itemsets using SiBlings and Subsets)for the following parts. Evaluations show adding subsets is more efficient in terms of lost itemsets.

Zoo Dataset

This dataset has 36 items and 101 transaction with average length of 16. Using minimum support count 70(%70), we obtain 23 frequent itemsets where 15 of which are not singleton. We evaluated the approach in (Ayav & Ergenc, 2015), HISB, HISB+S. Every 15 itemsets are selected as sensitive and hidden. In other words, we have 15 hiding scenarios. The average lost itemsets per scenario and the percentage of lost itemsets to all non-sensitive itemsets are given in Table 4.14.

Table 4.14 Comparison on zoo dataset

	AE15	HISB	HISB+S
Average Lost Itemsets	0.46	1.2	0.93
Lost Itemsets Rate	%2.1	%5.5	%4.3

Vote Dataset

This dataset has 48 items and 435 transactions with average length of 16. Using minimum support count 196(%45), we obtain 31 frequent itemsets where 12 of them are no singleton. We evaluated the approach in (Ayav & Ergenc, 2015), HISB, HISB+S. Every 12 itemsets are selected as sensitive and hidden. In other words, we have 12 hiding scenarios. The average lost itemsets per scenario and the percentage of lost itemsets to all non-sensitive itemsets are given in Table 4.15.

Table 4.15 Comparison on vote dataset

	AE15	HISB	HISB+S
Average Lost Itemsets	0	0.25	0.16
Lost Itemsets Rate	%0	%0.8	%0.5

Additional experiments show the effectiveness of the approaches in terms of lost itemsets. AE15 performs the best. This is what is expected since it generates more constraints and has control over all itemsets. HISB+S is the runner-up. Considering that datasets are small, the lost itemset rate is tolerable.

4.7.3 Comparison of Eliminating Prior Mining

In this section, we give a performance evaluation of the sibling itemset concept with two different approaches. We implemented two algorithms that differ from other by using frequent itemsets mined or not. In other words, sibling itemsets being found during hiding process or before hiding process with needs prior mining. These two are named HISB and HISBP respectively for ease of understanding during this part. Algorithms are tested with different parameters that are support count of sensitive itemsets, the number of sensitive itemsets, and the size of sensitive itemsets. We implemented the algorithms using Python. Constraints are solved using Minizinc (Nethercote et al, 2007). Implementations use Pymzn (Dragone, 2022) library to be able to invoke, run and gather results from the constraint solver. For the algorithm which uses frequent itemsets discovered prior to the hiding process, Python implementation (Borgelt, 2022) of the Eclat (Borgelt, 2003) algorithm is used for frequent itemset mining. All computational experiments are conducted on a PC running MS Windows 10 with Intel i7-6500U CPU and 16 GB RAM.

We evaluated the algorithms on three different datasets from (Goethals, 2022). Characteristics of these datasets are given in Table 4.16.

Table 4.16 Dataset properties of evaluation

Dataset Name	Number of Transactions	Average Transaction Length	Number of Items	Minimum support count	Number of Frequent itemsets
T10I4D100K	100000	10.10	870	500(%0.5)	1073
T40I10D100K	100000	39.60	942	500(%0.5)	1286037
Mushroom	8124	23.00	119	406(%5)	3755704

We experiment the algorithms using different hiding scenarios: hiding 1 2-itemset (HS_2.1), hiding 2 2-itemset(HS_2.2), hiding 3 2-itemset(HS_2.3), hiding 1 3-itemset(HS_3.1), hiding 2 3-itemset(HS_3.2), hiding 1 4-itemset(HS_4.1). The sensitive itemsets chosen have support counts close to minimum support count since those itemsets are more logically to be hidden and indistinguishable compared to the rest.

The results of our evaluation are given in Table 4.17. Columns representing dataset name, hiding scenario, and running time in seconds for algorithm HISB and algorithm HISBP. The last column gives time consumption when prior frequent itemset mining (FIM) is also added to the HISBP approach.

Table 4.17 Dataset properties of evaluation

Dataset Name	Hiding Scenario	Algorithm HISB (seconds)	Algorithm HISBP (seconds)	Algorithm HISBP + FIM (seconds)
T10I4D100K	HS_2.1	4.45	4.40	10.10
T10I4D100K	HS_2.2	5.35	5.14	10.84
T10I4D100K	HS_2.3	6.95	6.25	11.95
T10I4D100K	HS_3.1	5.14	5.03	10.73
T10I4D100K	HS_3.2	7.38	6.52	12.22
T10I4D100K	HS_4.1	7.76	7.15	12.85
T40I10D100K	HS_2.1	11,56	17.48	241.38
T40I10D100K	HS_2.2	17.61	22.09	245.99
T40I10D100K	HS_2.3	26.80	27.79	251.69
T40I10D100K	HS_3.1	18.28	20.05	243.95
T40I10D100K	HS_3.2	31.23	28.36	252.26
T40I10D100K	HS_4.1	19.69	20.09	243.99
Mushroom	HS_2.1	3.88	24.18	78.66
Mushroom	HS_2.2	6.72	27.81	82.29
Mushroom	HS_2.3	9.68	32.06	86.54
Mushroom	HS_3.1	4.40	25.45	79.93
Mushroom	HS_3.2	8.55	31.20	85.68
Mushroom	HS_4.1	11.34	31.18	85.66

If we compare the algorithms, it is clear that algorithm HISBP is slightly faster than algorithm HISB based on experiments on T10I4D100K dataset. The main reason behind this difference is the characteristic of the dataset. Frequent itemset mining on this dataset results in 1073 itemsets. However, if we include time for finding frequent itemsets, algorithm HISBP falls behind since the Eclat implementation we have used computes all frequent itemsets in 5.70 seconds. On the other hand, experiments on

T40I10D100K dataset show that algorithm HISB is faster. The main reason for this difference is that there are 1286037 frequent itemsets in this dataset, and searching sibling itemsets is not efficient enough. In addition, finding all frequent itemsets on this dataset is done in 223.90 seconds, and if we include this difference, then algorithms are not even comparable. Experiments on the Mushroom dataset reveal that eliminating prior mining is very advantageous when the dataset is very dense. Although this dataset has fewer items and transactions, the number of frequent itemsets for the given support threshold is 3755704. To add this, finding these itemsets costs 54.48 seconds. As a result, the elimination of prior mining for finding sibling itemsets is very efficient in terms of runtime.

CHAPTER 5

STUDY ON PERSONAL DATA AND ITS SENSITIVITY

5.1 Introduction

With the growing ability to process enormous volumes of data, we come across problems that we do not need to keep in mind earlier: privacy and processing of sensitive information. Therefore, developed systems for data analysis should comply with privacy concerns. By design and through access control mechanisms, such systems can be developed in awareness of privacy (Gurses et al., 2011). With regard to this, we surveyed to find awareness of people on the sensitivity of their personal data. Based on the collected data, we also conducted some analysis.

5.2 Questionnaire on Personal Data and Sensitivity Awareness

There are 25 questions. We can group questions in the questionnaire into six sets in order to obtain data about:

1. Non-specific personal information
2. Altitude to give misleading information
3. Change in privacy concerns in different situations
4. Loyalty cards and information provided
5. Concerns about information sharing among companies
6. Altitude to social networks and location information

The questionnaire was prepared using Google Forms (Yildiz, 2022). Submissions are recorded as transactions. Between 13.08.2020 and 30.10.2020, 171 transactions were recorded.

Google Forms gives us basic statistical results, in other words, a summary of answers. Although this is not our main objective, it would be beneficial to mention it. You can also find the summary for all questions as distribution charts in the appendix.

Distribution of responders' age (question 1) shows us that not distributing the questionnaire to mail groups and forums was a good decision for reaching a wide range of age groups without bias.

Question 2 shows good distribution of participants' sex. A nearly equal number of male and female participants responded to the questionnaire.

The difference in the distribution of answers for Questions 4 and 5 shows us that people are willing to give true information when they are using health services, and they do not usually hesitate. However, when we look at Questions 6, 7, and 8, we observe that people have different levels of sensitivity to private information in different departments of health services.

Questions from 10 to 15 show us that people are widely using loyalty cards, and they give sensitive information for usage. However, people are not comfortable when this information is shared with third parties. This can be observed in Questions 15 to 21. In addition, a tendency to give misleading information is observable when giving real information is not obligatory.

Responders of this questionnaire do not usually share sensitive information when using social network applications. It can be observed in questions 22 to 24. Question 25 shows us that location information provided to third parties is very annoying for people.

After the collection of answers, we combined question numbers and choice selections in order to obtain a dataset suitable for frequent itemset mining. The properties of the such dataset are in Table 5.1.

Table 5.1 Privacy awareness questionnaire dataset properties

Dataset Name	Number of Transactions	Average Transaction Length	Number of Items
privacy_quest_T171	171	23.8	113

5.3 Association Analysis on Survey Data

In the analysis of answers in terms of co-occurrence, we used the Apriori algorithm (Agrawal et al., 1994) implementation of Cristian Borgelt (Borgelt, 2022). The top five co-occurrence (frequent itemsets) is given in Table 5.2.

Table 5.2 Frequent itemsets for privacy awareness questionnaire

Itemset	Support(%)
13. Did you provide a phone number for any discount/loyalty card registration? **a)** Yes **11.** Did you provide first and last name information for any discount/loyalty card registration? **a)** Yes	68.4
16. Rate how uncomfortable it would be for you to be contacted by SMS or voice phone call for campaigns related to the products or services you have purchased before. When it is a different company/institution than the one I performed the transaction: **e)** 4(too much) **7.** Suppose you are waiting in line at the hospital. Your name and surname are displayed along with your serial number on the information screen. Please rate how much this situation disturbs you according to the unit you are in. Ophthalmology unit: **a)** 0(none)	62.5
5. Which one is valid for the information you enter in the hospital admission registration form, the preliminary information form about the condition of your disease, or the forms you are asked to fill in to monitor its progress over time? **a)** I always enter correct information without hesitation. **7.** Suppose you are waiting in line at the hospital. Your name and surname are displayed along with your serial number on the information screen. Please rate how much this situation disturbs you according to the unit you are in. Ophthalmology unit: **a)** 0(none)	61.4

Table 5.2 Continues

Itemset	Support(%)
11. Did you provide first and last name information for any discount/loyalty card registration? **a)** Yes **16.** Rate how uncomfortable it would be for you to be contacted by SMS or voice phone call for campaigns related to the products or services you have purchased before. When it is a different company/institution than the one I performed the transaction: **e)** 4(too much)	58.4
13. Did you provide a phone number for any discount/loyalty card registration? **16.** Rate how uncomfortable it would be for you to be contacted by SMS or voice phone call for campaigns related to the products or services you have purchased before. When it is a different company/institution than the one I performed the transaction: **e)** 4(too much)	57.8

In given frequent itemset analysis results, the most interesting one is {13a, 16e} with support of 57.8%. This itemset gives association rule 13a → 16e with a confidence of 83%. We observe that people who tend to give phone numbers for loyalty cards also find it annoying when a company other than they buy goods reaches them. In other words, people do not want third parties to reach their information.

5.4 Cluster Analysis on Survey Data

5.4.1 Determining the number of clusters:

Firstly, hierarchical agglomerative clustering is used. It is a bottom-up approach. It starts with small clusters and continues merging them to create larger clusters. Since we were not sure about the number of clusters, it was our first choice. Following, we obtained a dendrogram visualizing merges. This also served us to figure out the number of clusters we needed. It can be found by determining the largest vertical distance that doesn't intersect any of the other clusters. In the figure below, the results of clustering are depicted. The number of clusters in descending order of optimality are 2, 3, 4, and 5.

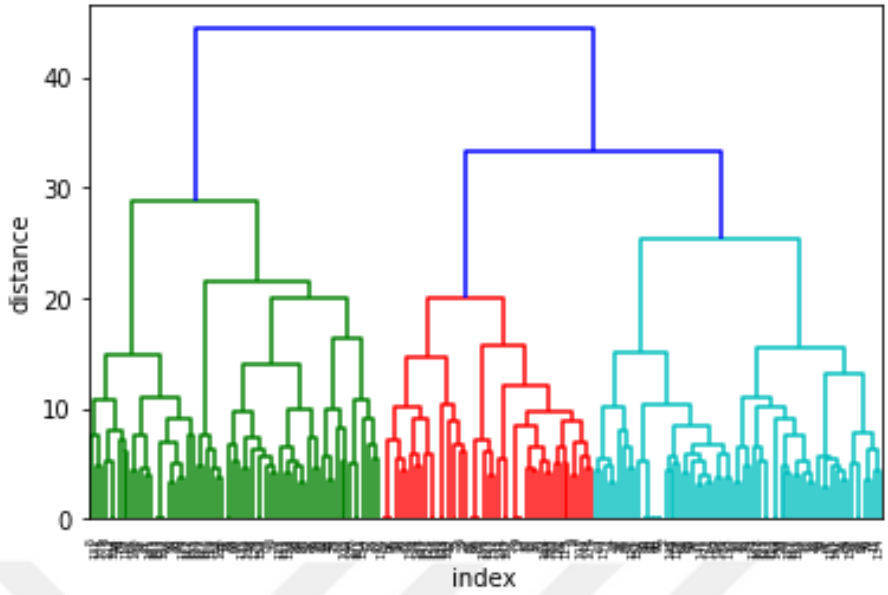


Figure 5.2 Agglomerative clustering dendrogram

Secondly, the elbow method is applied. With different k values, the k-means++ algorithm is executed, and the within-cluster sum of squared errors is recorded. We obtained similar observations, and the number of clusters was determined to be at most 5.

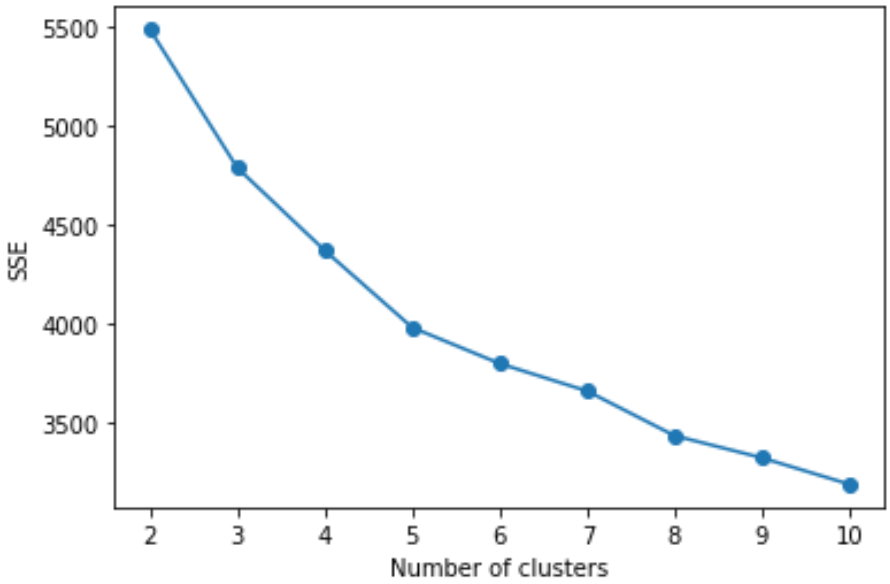


Figure 5.3 Elbow method

Another method is using the silhouette coefficient. It uses the mean of intra-cluster and the mean of the nearest cluster for each data point. It ranges from -1 to 1. A bigger value means better clusters since they are well apart from each other. The figure below gives the change of this coefficient over the number of clusters. By this method, we can say that 2, 3, and 5 are the number of clusters in descending order of optimality.

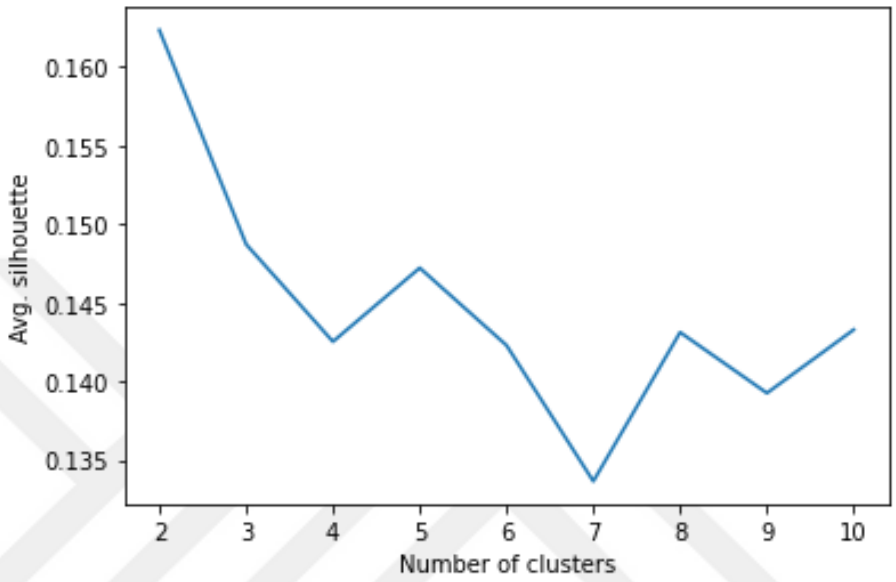


Figure 5.4 Silhouette Coeffcient

5.4.2 Cluster Results

Using the k-means++ algorithm (Vassilvitskii,& Arthur 2006) we clustered the dataset into 3 clusters. The results are given below.

Table 5.3 Cluster points

Question	Cluster0	Cluster1	Cluster2
1	d	c	e
2	b	a	b
3	e	e	e
4	b	b	a
5	a	a	a
6	a	a	a
7	a	a	a
8	a	a	a
9	c	a	a
10	c	a	c
11	a	z	a
12	c	z	c
13	a	z	a
14	e	z	e
15	e	c	d
16	e	e	e
17	a	b	a
18	c	c	c
19	e	e	c
20	e	e	e
21	f	f	e
22	a	a	a
24	f	a	a
25	e	e	e

We can interpret the results as follows:

- Cluster2: People above 50 are likely to enter correct values to forms
- Cluster1: People who do not have loyalty cards are not likely to click advertisement links and do not share anything on social media
- Cluster0: These people are not comfortable with information shared with third parties and have concerns about entering correct values into forms

Classification on Assigned Clusters

We assign clusters as classes to instances and run three different classification algorithms available in Weka (Hall et al., 2009). These are J48 (Quinlan, 1993), Random Forest (Breiman, 2001), IBk (Aha et al., 1991). Correctly classified instances are given in the table below.

Table 5.4 Classification correctness of clusters

Algorithm	**% of Correct Classes**
J48	77.7
J48(unpruned)	81.8
Random Forest	86.5
IBk	84.7

The classification on assigned clusters shows the success of cluster analysis on survey data. Pruned tree constructed by J48 algorithm in Weka is given below.

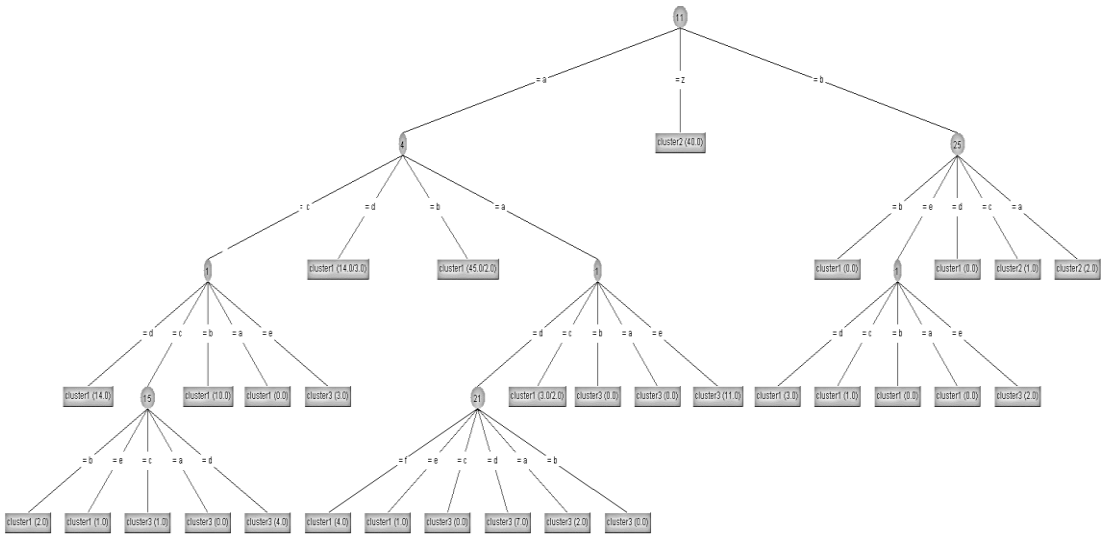


Figure 5.5 J48 Decision Tree for Classification of Clusters

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Firstly, this thesis presents a methodology for hiding sensitive itemsets in transactional datasets. We focused on reducing the number of constraints for exact itemset hiding. Using sibling itemset notion and defining relaxation variables for constraints, we benefited from the exact nature of algorithms to obtain an ideal solution or minimally affected dataset. We showed that sibling itemsets are an efficient solution for reducing constraints for exact approaches. Given a comparison with a reference algorithm, we also discuss the need for prior computation of frequent itemsets. Experiments revealed that eliminating prior mining of frequent itemsets on the dataset combined with a sibling itemset approach is time-efficient where side effects such as lost itemsets are tolerable. Our approach is especially applicable where prior mining of frequent itemsets is costly. This is also valid for frequently updated databases. Therefore, we can say that skipping prior mining while using constraints is one of the most important contributions of our approach. Additionally, using fewer constraints makes our approach even better in terms of runtime. Moreover, we added relaxation variables to make our approach more efficient when initial constraints cause a CSP that is not feasible. Although it is observed not to be common, it may result in additional runtime since the constraint solver needs to be run more than once. It can be concluded that our methodology serves an exact approach with fewer constraints so that the hiding process consumes less time. The main findings from the technical part of the thesis study can be listed as follows:

- The sibling itemset concept is efficient in reducing constraints for CSP based itemset hiding.
- The sibling itemset concept combined with the elimination of prior mining is efficient in terms of runtime.
- Using relaxation variables for constraints does not have much effect on runtime since it is rarely observed.

Secondly, we surveyed the privacy awareness of people. It is seen that collected data has good distribution in terms of sex and age, so applicable for analysis without bias. Association and cluster analyses are made. The main findings from this part of the thesis are as follows:

- People over 50 are likely to enter correct values into forms
- People who do not have loyalty cards are not likely to click advertisement links and not share anything on social media
- Some people are not comfortable with information shared with third parties and have concerns about entering correct values into forms
- People are not against information sharing. Instead, they are against third parties reaching their personal data

6.2 Future Work

In this thesis study, an approach for itemset hiding is presented, which uses item deletion. The proposed approach may be extended to add items in order to solve the problem of lost itemsets.

Since the proposed approach is more efficient for databases updated frequently by skipping prior mining of itemsets, dynamic hiding capabilities can be studied.

Some privacy problems may need a different level of privacy level for different itemsets. For such situations, some approaches are proposed to hide itemsets with multiple support thresholds. The approach proposed in this study can be modified for such problems.

The items of the database in this study are considered equally important as it is in frequent itemset mining. For situations where items of the database have different importance, researchers are studying utility itemset mining. The applicability of the proposed itemset hiding approach for such databases and required modifications can be studied.

Another study in this thesis is a survey, and the data collected may be analyzed further with different aspects and techniques. In addition, the dataset obtained and the results of the analysis need interpretations. It can be formatted and published as public for the usage of others.

REFERENCES

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, 207–216. https://doi.org/10.1145/170035.170072

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *20th Int. Conf. Very Large Data Bases, VLDB*, *1215*, 487–499.

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, *6*(1), 37–66. https://doi.org/10.1007/bf00153759

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, *46*(3), 175–185. https://doi.org/10.1080/00031305.1992.10475879

Amanowicz, M., & Jankowski, D. (2021). Detection and classification of malicious flows in software-defined networks using data mining techniques. *Sensors*, *21*(9), Article 9. https://doi.org/10.3390/s21092972

Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1027–1035.

Atallah, M., Bertino, E., Elmagarmid, A., Ibrahim, M., & Verykios, V. (1999). Disclosure limitation of sensitive rules. *Proceedings 1999 Workshop on Knowledge and Data Engineering Exchange (KDEX'99) (Cat. No.PR00453)*, 45–52. https://doi.org/10.1109/KDEX.1999.836532

Ayav, T., & Ergenc, B. (2015). Full-exact approach for frequent itemset hiding: *International Journal of Data Warehousing and Mining*, *11*(4), 49–63. https://doi.org/10.4018/ijdwm.2015100103

Bertino, E., Fovino, I. N., & Provenza, L. P. (2005). A framework for evaluating privacy preserving data mining algorithms*. *Data Mining and Knowledge Discovery*, *11*(2), 121–154. https://doi.org/10.1007/s10618-005-0006-6

Borgelt, C. (2003). Efficient implementations of apriori and eclat. *FIMI'03: Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, 9.

Borgelt, C. (2022, August 31). *Christian Borgelt's Web Pages*. https://borgelt.net/fpm.html

Breiman, L. (1984). *Classification and regression trees*. Routledge. https://doi.org/10.1201/9781315139470

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Dasseni, E., Verykios, V. S., Elmagarmid, A. K., & Bertino, E. (2001). Hiding association rules by using confidence and support. In I. S. Moskowitz (Ed.), *International Workshop on Information Hiding* (Vol. 2137, pp. 369–383). Springer. https://doi.org/10.1007/3-540-45496-9_27

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning*, *29*(2), 103-130.

Dragone, P. (2022, August 31). *PyMzn—PyMzn 0.17.0 documentation*. https://paolodragone.com/pymzn

Dunham, M. H. (2003). Data mining introductory and advanced topics. Prentice Hall/Pearson Education.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *96*(34), 226–231.

Feng, K., & Tian, J. (2021). Forecasting reference evapotranspiration using data mining and limited climatic data. *European Journal of Remote Sensing*, *54*(sup2), 363–371. https://doi.org/10.1080/22797254.2020.1801355

Gkoulalas-Divanis, A., & Verykios, V. S. (2006). An integer programming approach for frequent itemset hiding. *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, 748–757. https://doi.org/10.1145/1183614.1183721

Gkoulalas-Divanis, A., & Verykios, V. S. (2009). Hiding sensitive knowledge without side effects. *Knowledge and Information Systems*, *20*(3), 263–299. https://doi.org/10.1007/s10115-008-0178-7

Gkoulalas-Divanis, A., & Verykios, V. S. (2010). Association rule hiding for data mining (Vol. 41). Springer US. https://doi.org/10.1007/978-1-4419-6569-1

Goethals, B. (2022, August 31). Frequent Itemset Mining Dataset Repository. http://fimi.uantwerpen.be/data/

Guanling Lee, Chien-Yu Chang, & Chen, A. L. P. (2004). Hiding sensitive patterns in association rules mining. *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.*, 424–429. https://doi.org/10.1109/CMPSAC.2004.1342874

Gürses, S., Troncoso, C., & Diaz, C. (2011). Engineering privacy by design. *Computers, Privacy & Data Protection*, *14*(3), 25.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10–18. https://doi.org/10.1145/1656274.1656278

Han, J., & Kamber, M. (2006). *Data mining: concepts and techniques* (2nd ed., Issue March). Morgan Kaufmann.

Han, J., & Kamber, M. (2012). *Data mining: concepts and techniques* (3rd ed). Elsevier.

Han, J., Pei, J., Yin, Y., & Mao, R. (2004). Mining frequent patterns without candidate generation: a frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, *8*(1), 53–87. https://doi.org/10.1023/B:DAMI.0000005258.31418.83

Hong, J.-W., & Park, S.-B. (2019). The Identification of Marketing Performance Using Text Mining of Airline Review Data. *Mobile Information Systems*, *2019*, 1–8. https://doi.org/10.1155/2019/1790429

Hopfield, J. J., & Tank, D. W. (1985). "Neural" computation of decisions in optimization problems. *Biological Cybernetics*, *52*(3), 141–152. https://doi.org/10.1007/BF00339943

Khuda Bux, N., Lu, M., Wang, J., Hussain, S., & Aljeroudi, Y. (2018). Efficient association rules hiding using genetic algorithms. *Symmetry*, *10*(11), 576. https://doi.org/10.3390/sym10110576

Lefkir, M., Nouioua, F., & Fournier-Viger, P. (2022). Hiding sensitive frequent itemsets by item removal via two-level multi-objective optimization. *Applied Intelligence*. https://doi.org/10.1007/s10489-022-03808-6

Lin, C.-W., Hong, T.-P., & Hsu, H.-C. (2014). Reducing side effects of hiding sensitive itemsets in privacy preserving data mining. *The Scientific World Journal*, *2014*, e235837. https://doi.org/10.1155/2014/235837

Lin, J. C.-W., Liu, Q., Fournier-Viger, P., Hong, T.-P., Voznak, M., & Zhan, J. (2016). A sanitization approach for hiding sensitive itemsets based on particle swarm optimization. *Engineering Applications of Artificial Intelligence*, *53*, 1–18. https://doi.org/10.1016/j.engappai.2016.03.007

Liu, L., & Özsu, M. T. (Eds.). (2018). Encyclopedia of database systems. Springer New York. https://doi.org/10.1007/978-1-4614-8265-9

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129–137. https://doi.org/10.1109/TIT.1982.1056489

Mannila, H., & Toivonen, H. (1997). Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, *1*(3), 241–258.

Mendes, R., & Vilela, J. P. (2017). Privacy-preserving data mining: methods, metrics, and applications. *IEEE Access*, *5*, 10562–10582. https://doi.org/10.1109/ACCESS.2017.2706947

Menon, S., Sarkar, S., & Mukherjee, S. (2005). Maximizing accuracy of shared databases when concealing sensitive patterns. *Information Systems Research*, *16*(3), 256–270. https://doi.org/10.1287/isre.1050.0056

Moustakides, G. V., & Verykios, V. S. (2008). A maxmin approach for hiding frequent itemsets. *Data & Knowledge Engineering*, *65*(1), 75–89. https://doi.org/10.1016/j.datak.2007.06.012

Neto, C., Brito, M., Lopes, V., Peixoto, H., Abelha, A., & Machado, J. (2019). Application of data mining for the prediction of mortality and occurrence of complications for gastric cancer patients. *Entropy*, *21*(12), Article 12. https://doi.org/10.3390/e21121163

Oliveira, S. R. M., & Zaïane, O. R. (2002). Privacy preserving frequent itemset mining. *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining - Volume 14*, 43–54.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106. https://doi.org/10.1007/BF00116251

Quinlan, J. R. (1993). C4.5: programs for machine learning. Elsevier.

Quoc Le, H., Arch-int, S., & Arch-int, N. (2013). Association rule hiding based on intersection lattice. *Mathematical Problems in Engineering*, *2013*, 1–11. https://doi.org/10.1155/2013/210405

Raja, K., Patrick, M., Gao, Y., Madu, D., Yang, Y., & Tsoi, L. C. (2017). A review of recent advancement in integrating omics data with literature mining towards

biomedical discoveries. *International Journal of Genomics*, *2017*, 1–10. https://doi.org/10.1155/2017/6213474

Sánchez-Aguayo, M., Urquiza-Aguiar, L., & Estrada-Jiménez, J. (2022). Predictive fraud analysis applying the fraud triangle theory through data mining techniques. *Applied Sciences*, *12*(7), Article 7. https://doi.org/10.3390/app12073382

Saygin, Y., Verykios, V. S., & Clifton, C. (2001). Using unknowns to prevent discovery of association rules. *ACM SIGMOD Record*, *30*(4), 45–54. https://doi.org/10.1145/604264.604271

Sun, X., & Yu, P. S. (2007). Hiding sensitive frequent itemsets by a border-based approach. *Journal of Computing Science and Engineering*, *1*(1), 74–94. https://doi.org/10.5626/JCSE.2007.1.1.074

Verykios, V. S., Elmagarmid, A. K., Bertino, E., Saygin, Y., & Dasseni, E. (2004). Association rule hiding. *IEEE Transactions on Knowledge and Data Engineering*, *16*(4), 434–447. https://doi.org/10.1109/TKDE.2004.1269668

Wang, S.-L., & Jafari, A. (2005). Using unknowns for hiding sensitive predictive association rules. *IRI -2005 IEEE International Conference on Information Reuse and Integration, Conf, 2005.*, 223–228. https://doi.org/10.1109/IRI-05.2005.1506477

Yildiz, B. (2022, August 31). *Kisisel Veri ve Hassasiyeti Farkındalık Anketi* https://forms.gle/BZXAsnXoDJh3S5Mv5

Yildiz, B., Kut, A., & Yilmaz, R. (2022). Hiding sensitive itemsets using sibling itemset constraints. *Symmetry*, *14*(7), 1453. https://doi.org/10.3390/sym14071453

Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1997). New algorithms for fast discovery of association rules. In *KDD* (Vol. 97, pp. 283-286)..

Zhang, L., Wang, W., & Zhang, Y. (2019). Privacy preserving association rule mining: taxonomy, techniques, and metrics. *IEEE Access*, *7*, 45032–45047. https://doi.org/10.1109/ACCESS.2019.2908452

APPENDICES

Summary results for each question of survey:

1. What is your age range? 171 responses

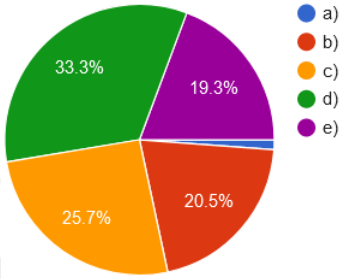
a) 2 less than →20

b) Between 20-30 →35

c) Between 30-40 →44

d) Between 40-50 →57

e) 50 and above →33

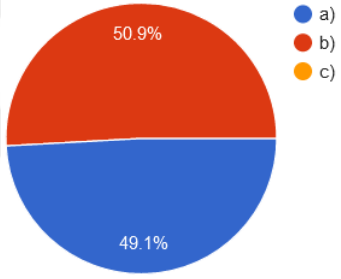


2. What is your gender? 171 responses

a) Male →84

b) Woman →87

c) Other →0



3. What is your graduation status? 171 responses

a) Primary School →0

b) Secondary School →7

c) High School →19

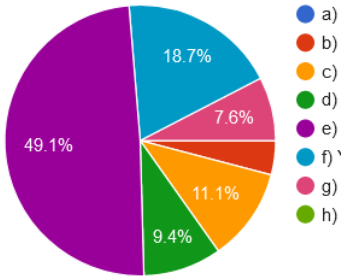
d) College →16

e) University →84

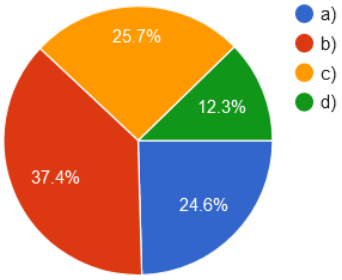
f) Graduate →32

g) PhD →13

h) None →0



4. Which one is valid for the information you enter in forms such as discount card form, shopping site membership form? 171 responses



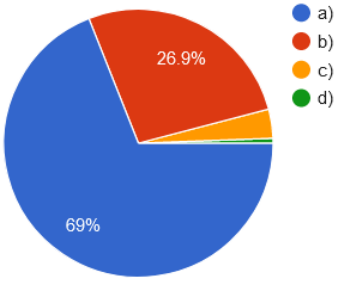
a) I always enter correct information without hesitation. →42

b) I have always given correct information, even if there are situations where I hesitate. →64

c) I have knowingly and willingly given incomplete information at least once. →44

d) I have knowingly and willfully given false information at least once. →21

5. Which one is valid for the information you enter in the hospital admission registration form, the preliminary information form about the condition of your disease, or the forms you are asked to fill in to monitor its progress over time? 171 responses



a) I always enter correct information without hesitation. →118

b) I have always given correct information, even if there are situations where I hesitate. →46

c) I have knowingly and willingly given incomplete information at least once. →6

d) I have knowingly and willfully given false information at least once. →1

Answer questions 6, 7 and 8 with this background information: Suppose you are waiting in line at the hospital. Your name and surname are displayed along with your serial number on the information screen. Please rate

how much this situation disturbs you according to the unit you are in. (0 none, 4 most)

6. Mental health and diseases unit: 171 responses

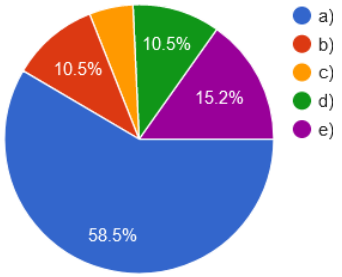
a) 0 →100

b) 1 →18

c) 2 →9

d) 3 →18

f) 4 →26



7. Eye diseases unit: 171 responses

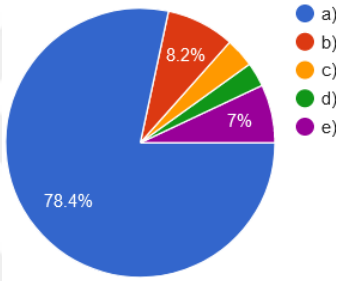
a) 0 →134

b) 1 →14

c) 2 →6

d) 3 →5

f) 4 →12



8. Venereal diseases unit: 171 responses

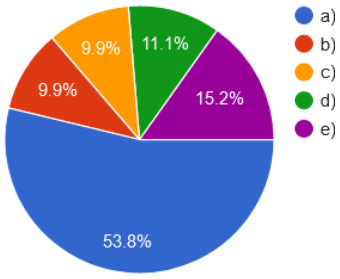
a) 0 →92

b) 1 →17

c) 2 →17

d) 3 →19

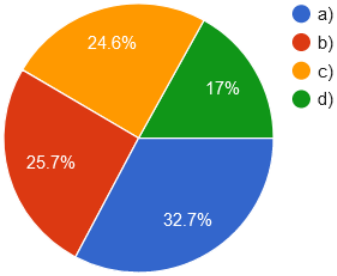
f) 4 →26



9. What is true about the information you provided for general trending surveys such as street surveys, satisfaction surveys? 171 responses

a) I always enter correct information without hesitation. →56

b) I have always given correct information, even if there are situations where I hesitate. →44



c) I have knowingly and willingly given incomplete information at least once. →42

d) I have knowingly and willfully given false information at least once. →29

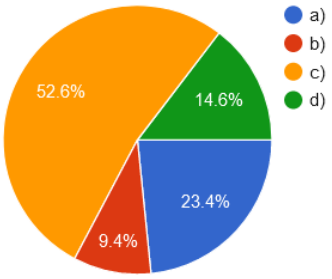
10. Do you use a discount/loyalty card for your purchases?171 responses

a) I have never registered a card. →40

b) I have a card but I do not use it. →16

c) I use the discount quite often. →90

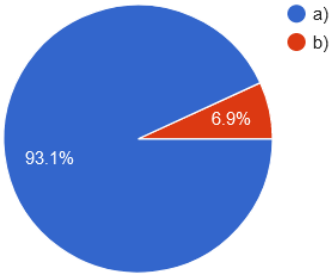
d) I try to use it as often as possible. →25



11. Did you provide name and surname information for any discount/loyalty card registration?131 responses

a) Yes →122

b) No →9



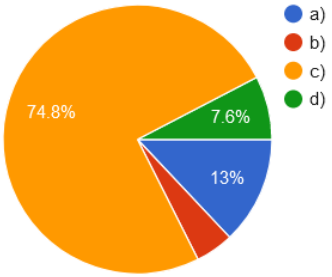
12. What was the most comprehensive date of birth information you provided for any discount/loyalty card registration?131 responses

a) Year →17

b) Month and year →6

c) Day, month and year →98

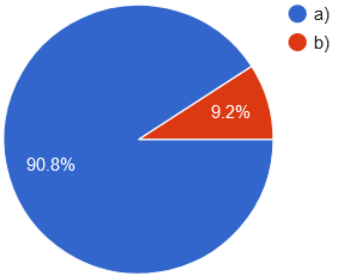
d) None →10



13. Did you provide a phone number for any discount/loyalty card registration?131 responses

a) Yes →119

b) No →12



14. How detailed was the most comprehensive address information you provided for any discount/loyalty card registration?131 responses

a) Province →17

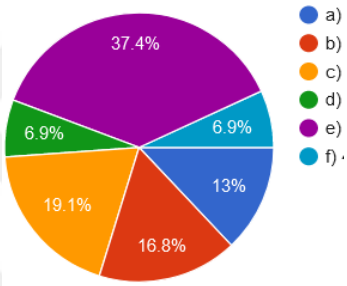
b) District →22

c) Neighborhood/District →25

d) Street →9

e) House/Building No →49

f) None →9



Answer questions 15 and 16 with this preliminary information: Rate how uncomfortable it would be for you to be contacted by SMS or voice phone call for campaigns related to products or services you have purchased before. (0 none, 4 most)

15. If the company/institution I perform the transaction: 171 responses

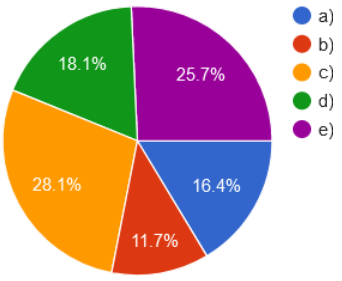
a) 0 →28

b) 1 →20

c) 2 →48

d) 3 →31

f) 4 →44



16. If it is a different company/institution than the one I performed the transaction: 171 responses

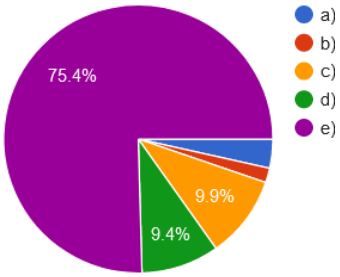
a) 0 →6

b) 1 →3

c) 2 →17

d) 3 →16

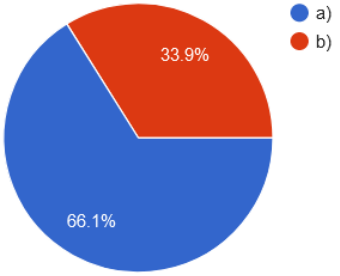
e) 4 →129



17. Did you voluntarily click on an advertisement link while surfing the Internet?171 responses

a) Yes →113

b) No →58



Answer questions 18, 19, 20 and 21 with this preliminary information: Rate how uncomfortable it would be for you to see advertisements for the products/services you have purchased or are interested in. (0 none, 4 most)

18. While navigating the site where I performed the transaction:171 responses

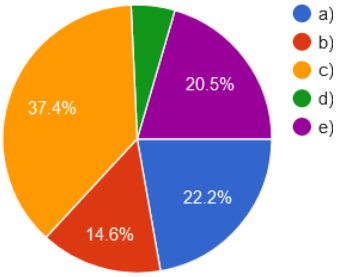
a) 0 →38

b) 1 →25

c) 2 →64

d) 3 →9

d) 4 →35



19. Browsing a news site:171 responses

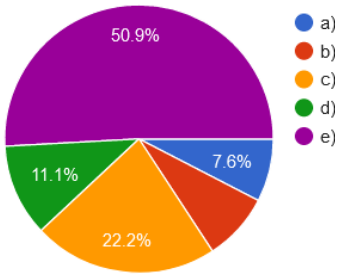
a) 0 →13

b) 1 →14

c) 2 →38

d) 3 →19

f) 4 →87



20. When using a phone app: 171 responses

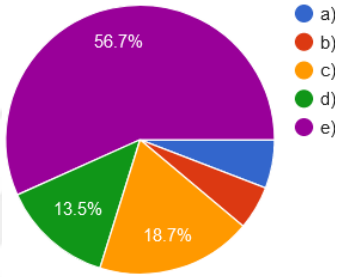
a) 0 →10

b) 1 →9

c) 2 →32

d) 3 →23

f) 4 →97



21. When using social networking (Facebook, Instagram and similar) applications: 171 responses

a) I do not use social networks. →12

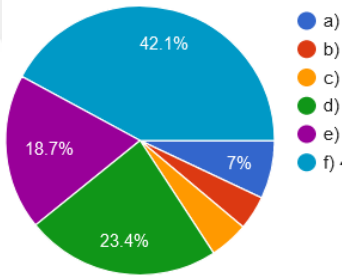
b) 0 →7

c) 1 →8

d) 2 →40

d) 3 →32

f) 4 →72



22. Rate your tendency to provide location information in social networking posts. (0 none, 4 most)159 responses

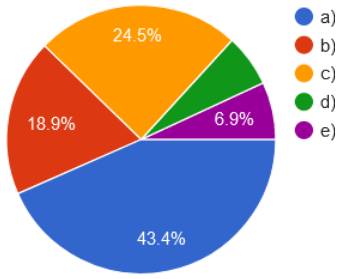
a) 0 →69

b) 1 →30

c) 2 →39

d) 3 →10

f) 4 →11



23. For what purposes do you use location information in your social network shares? (You can mark more than one option.)159 responses

a) To take advantage of the campaigns. →18

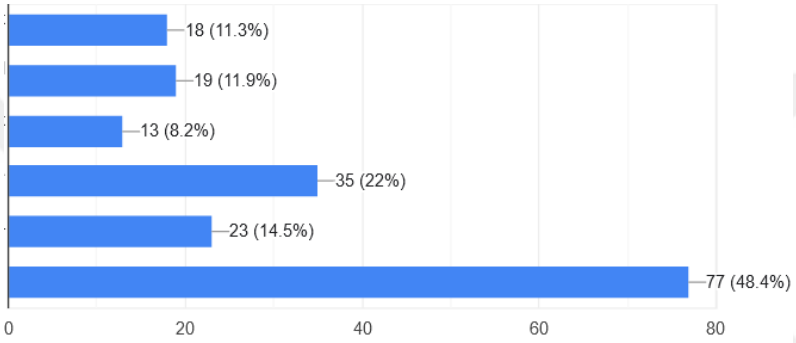
b) For social interaction with those in that position. →19

c) To map my posts. →13

d) To remember later where I have been →35

e) Other →23

f) I do not share location information →77



24. Rate your tendency to hide things that can identify you such as license plate, phone number, ID number in your social network posts. (0 none, 4 most)159 responses

a) I don't share anything. →64

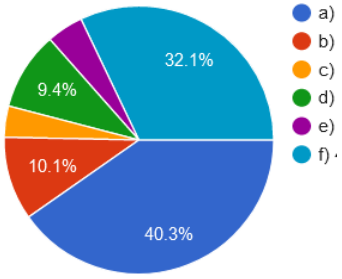
b) 0 →16

c) 1 →6

d) 2 →15

d) 3 →7

f) 4 →51



25. Rate how uncomfortable it would be for you to see advertisements on your phone related to your location. (0 none, 4 most)171 responses

a) 0 →13

b) 1 →17

c) 2 →29

d) 3 →29

f) 4 →83