Expert Systems With Applications 96 (2018) 406–426

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

Data sanitization in association rule mining: An analytical review

Akbar Telikani, Asadollah Shahbahrami*

Department of Computer Engineering, Faculty of Engineering, University of Guilan, Rasht, Iran

ARTICLE INFO

Article history:
Received 20 March 2017
Revised 21 October 2017
Accepted 22 October 2017
Available online 24 October 2017

Keywords:
Privacy preserving in data mining
Association rule mining
Association rule hiding
Data sanitization

ABSTRACT

Association rule hiding is the process of transforming a transaction database into a sanitized version to protect sensitive knowledge and patterns. The challenge is to minimize the side effects on the sanitized database. Many different sanitization algorithms have been proposed to reach this purpose. This article presents a structured analysis and categorization of the existing challenges and directions for state-of-the-art sanitization algorithms, with highlighting about their characteristics. Fifty-four scientific algorithms, primarily spanning the period 2001–2017, were analyzed and investigated in terms of four aspects including hiding strategy, sanitization technique, sanitization approach, and selection method. In terms of results and findings, this review showed that (i) in comparison to other aspects of sanitization algorithms, the transaction and item selection methods more significantly influence the optimality of hiding process, (ii) blocking technique increases the disclosure risk while distortion technique is better in knowledge protection field, and transaction deletion/insertion technique is a new direction, (iii) heuristic-based algorithms have attracted more attention than other algorithms, especially in the context of hiding the association rules, (iv) a new trend is to use evolutionary paradigm for knowledge hiding that is often integrated with the transaction deletion/insertion technique, and (V) hiding the association rules introduces more challenges than hiding the frequent itemsets in terms of the determination of strategy and formulation of the selection method. This study aims to help researchers and database administrators find recent developments in association rule hiding.

1. Introduction

Data sharing may lead to the leakage of sensitive information related to the businesses' competitive edge or the individuals' privacy (Verykios et al., 2004a). The policies and guidelines in data publishing cannot protect this information because they rely on the publication of specified types of data and on agreements on the use of published data (Fung et al., 2010). Thus, before releasing the data, sensitive information is protected through database modification (Divanis & Verykios, 2009b). Protection methods can be classified into two main categories, (i) data hiding and (ii) knowledge hiding. Data hiding methods modify sensitive raw data using randomization techniques (Agrawal & Srikant, 2000; Evfimievski et al., 2004; Lin & Liu, 2007; Rizvi & Haritsa, 2002; Lin & Cheng, 2009) or modify quasi-identifiers using anonymization techniques to obscure the record owner (Samarati, 2001; Sweeney, 2002; Hajian et al., 2014), irrespective of the kind of analysis that is performed by the third party (Prakash & Singaravel, 2015). The quasi-identifiers attributes are those that cannot potentially identify record owner alone, but if they are combined together, might unambiguously identify the entity such as age and zip code (Fung et al., 2010; Hajian et al., 2014). In the literature, these methods are known as "Privacy-Preserving Data Publishing (PPDP)" (Fung et al., 2010).

The knowledge hiding methods focus on protecting the sensitive data mining results (Divanis & Verykios, 2010). This category is denoted as Privacy-Preserving Data Mining (PPDM). The privacy threats caused by data mining results were first introduced by O' Leary (1991, 1995). Then, Clifton and Marks (1996) presented some data-obscuring strategies to prohibit inference and discovery of sensitive knowledge. The PPDM can be applied in different data mining tasks such as association rule mining, clustering, and classification. Privacy preserving association rule mining is concerned with the sanitization of data leading to the disclosure of confidential and private knowledge (Divanis & Verykios, 2010). It is known as association rule hiding/data sanitization. In privacy preserving clustering, the cluster center is changed by distorting the confidential attributes. Oliveira and Zaiane (2004) perturbed confidential numerical attributes using geometric data transformation to meet privacy protection in clustering analysis, notably on partition-based and hierarchical clustering. Oliveira and Zaiane (2010) also applied the rotation-based transformation to preserve the privacy of information independently of any clus-

* Corresponding author.
E-mail address: shahbahrami@guilan.ac.ir (A. Shahbahrami).

https://doi.org/10.1016/j.eswa.2017.10.048
0957-4174/

tering algorithm. Privacy preserving classification approaches downgrade the effectiveness of classifiers such that the classifiers do not reveal any sensitive knowledge. Some of the PPDM techniques used in classification rule and decision tree applications were discussed in Chang and Moskowitz (1998) and Moskowitz and Chang (2000). Secure multi-party computation (Yao, 1982) is the most common cryptographic-based technique for privacy preserving distributed data mining (Lindell & Pinkas, 2009). Clifton et al. (2002) presented some techniques such as secure sum, the secure set union, secure size of set intersection, and scalar product that are useful for many data mining tasks. In recent years, the term "PPDP" has sometimes evolved to cover many PPDM research problems, even though they are not exactly the same.

The association rule hiding is one of the main research areas in PPDM that was suggested for the first time by Atallah et al. (1999). To illustrate the need to preserve sensitive association rules, a publicly released record of a bookstore's transactions reveals that people who bought a book titled "Romeo and Juliet" in the last month also bought a book titled "Free Alaska" . Alice meets Bob reading "Romeo and Juliet", and learns that he bought it during the last month. Thus, she makes an inference that violates Bob's privacy concerning his political persuasions (Cao et al., 2010). Similar motivating examples for association rule hiding are discussed in Clifton and Marks (1996), Divanis and Verykios, (2010), Oliveira and Zaiane (2002), and Sun and Yu (2007). The association rule hiding process sanitizes transactions to decrease the confidence/support of sensitive patterns below the predefined thresholds (Divanis & Verykios, 2010). This process produces some side effects on the sanitized database so that some non-sensitive patterns are lost or new patterns may be introduced. A sanitization solution that hides all sensitive knowledge and also produces no side effects is known as an "optimal solution". It is important to note that the problem of finding an optimal data sanitization is an NP-hard problem (Atallah et al., 1999). In the past two decades, several algorithms have been proposed to find this solution for hiding sensitive information.

Association rule hiding has been the topic of a number of surveys and review articles, as well as books, where the goal was to collect and classify association rule hiding algorithms. Verykios and Divanis (2008) and Verykios (2013) have briefly overviewed association rule hiding and presented a taxonomy of important sample of algorithms. Divanis and Verykios (2010) provided an extensive description of the main characteristics of sanitization algorithms. All existing surveys focus on providing a classification of association rule hiding algorithms based on the underlying approach adopted by each algorithm. They classified algorithms proposed from 2001 to 2009 into three approach-based categories, including heuristic, border, and exact. Besides, no research work, to the best of our knowledge, has been conducted to analyze the contributing factors on the utility of sanitization process, while the study of the impact and importance of these factors in data sanitization process is a very significant issue for both researchers and database administrators.

In this survey, we attempt to provide an analytical review of major directions in association rule hiding and present our own insights into this topic. Unlike other surveys that describe the algorithms in terms of an approach-based classification, our research does not intend to provide a detailed description of association rule hiding algorithms because some decent surveys already exist. The main objective is to introduce new concepts and trends about different perspectives of data sanitization process.

The paper has three objectives: the first is to review the up-to-date sanitization algorithms to provide a reference point for researchers and database administrators. The second is to analyze various aspects of association rule hiding with a broad discussion about their features, outstanding advantages, and disadvantages. The third goal is to present directions and trends in association rule hiding by introducing a statistical review of the application of sanitization aspects in the identified algorithms.

The rest of the paper is organized as follows: Section 2 presents the statement of the problem and the notations used in this study. Section 3 describes the association rule hiding process, and Section 4 introduces its side effects using an illustrative example. Section 5 describes data sanitization algorithms collected from 2001 to 2017 in chronological order. Section 6 analyzes the current directions in data sanitization for hiding sensitive knowledge. In Section 7, a discussion and statistical analysis of the collected algorithms is provided, while highlighting key benefits and limitations. Section 8 presents the measures and standard datasets for assessing the performance of sanitization algorithms. Finally, Section 9 offers conclusions and directions for future research.

2. Problem statement

Association rule mining is one of the most important data mining techniques, which was first introduced by Agrawal et al. (1993). Association rule mining algorithms are classified into two categories: *level-wise* and *pattern-growth*. Eclat (Zaki, 2000) and Apriori (Agrawal & Srikant, 1994) algorithms have been designed to mine the association rules in a level-wise way. The Apriori uses a breadth-first search to count the support of itemsets while the Eclat uses a depth-first search using intersection set. Like the Eclat algorithm, FP-growth algorithm (Han et al., 2000) performs a depth-first search without candidate creation by adopting "divide and conquer" approach (Sohrabi & Roshani, 2017). Instead of counting the support of a candidate set using the intersection-based approach, it uses frequent-pattern tree technique (Han et al., 2000) to store all transactions of the database in a trie based structure (Han et al., 2004). The basic concepts of association rule mining are defined as follows:

Let $I=\{i_1, i_2,\ldots,i_n\}$ is a set of n distinct items in database (D), $D=(t_1, t_2,\ldots, t_m)$ on I is a finite set of transactions. Each transaction t is a set of items in I, such that $t \subseteq I$. An association rule is expressed as $X \Longrightarrow Y$, such that $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \emptyset$. In order to discover an association rule, first, its generating itemset is extracted based on *support* criterion, denoted by α. Then, the rule is derived from the generating itemset based on *confidence* criterion, denoted by β. The support of the rule $X \Longrightarrow Y$ is the percentage of transactions that contain both X and Y, which is calculated using below equation

$$\alpha(X \Rightarrow Y) = \frac{|X \cup Y|}{m} \tag{1}$$

In this equation, m is the number of transactions in D and $|X \cup Y|$ is the number of transactions that include both itemsets X and Y. If support of the rule is above the minimum support threshold, denoted by $\alpha_{\min}$, then its generating itemset is called *frequent itemset* (Zeng et al., 2015). The confidence of a rule is the proportion of the transactions that contain X also contain Y the confidence is calculated using below equation.

$$\beta(X \Rightarrow Y) = \frac{|X \cup Y|}{|X|} \tag{2}$$

Where $|X|$ is the number of transactions that include itemset X. A rule is strong when its confidence is above the minimum confidence threshold, donated by $\beta_{\min}$ (Hai et al., 2013a).

Based on association rule mining property, a sensitive rule reveals the privacy when its support is greater than the $\alpha_{\min}$, or its confidence is higher than the $\beta_{\min}$. Therefore, in order to hide a sensitive rule, it is required to reduce its support or confidence below the minimum thresholds so that the rule cannot be discovered from the sanitized database. Briefly, association rule hiding can be

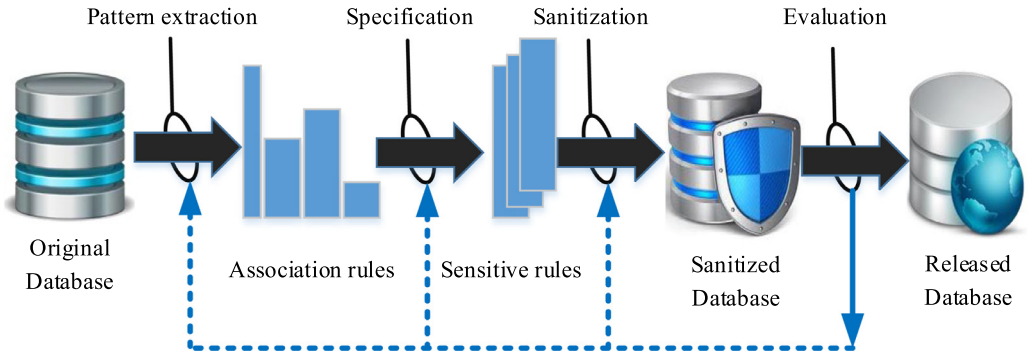


Fig. 1. Generic framework for association rule hiding process.

stated as follows: given a transaction database, a set of meaningful patterns that are mined from the original database, and a subset of sensitive patterns included in the mined patterns. We want to transform the database into a sanitized database in such a way that all sensitive patterns are hidden, while non-sensitive patterns can still be mined. Actually, association rule hiding acts in a different manner than the association rule mining. The objective of association rule hiding is to degrade the importance of sensitive patterns to a degree that they become uninteresting from the perspective of the data mining algorithms, while the objective of association rule mining is to extract unknown and interesting patterns from the transaction database.

3. Association rule hiding process

In association rule hiding process, the support and confidence thresholds are considered as *sensitivity level*. If support/confidence of a strong and frequent rule are above a certain sensitivity level, the hiding process should be applied in such a way that the frequency or the strength of the rule is reduced. This process contains four steps including pattern extraction, specification, sanitization, and evaluation, which are depicted in Fig. 1.

Step 1: *Pattern extraction,* a set of frequent itemsets or association rules are mined from the original database by using an association rule mining algorithm.

Step 2: *Specification*, some patterns or items that violate the privacy are specified by the user as sensitive. This step is different for the collaborative recommendation (Lin et al., 2002; O'Mahony et al., 2004) and predictive (Li et al., 2001) association rules. The sensitive predictive association rules and collaborative recommendation rules are the rules that contain sensitive items on the left-hand side and right-hand side of the rules respectively. For such rules, sensitive items are identified without pre-mining and selection of hidden rules. Therefore, the hiding process is integrated into the pre-process of finding these hidden rules (Wang et al., 2007a, 2007b, 2008; Wang & Jafari, 2005; Wang, 2009).

Step 3: *Sanitization*, in this step, the database is sanitized using a sanitization algorithm for the hiding of sensitive patterns. Applying an optimal algorithm reduces the side effects on the sanitized database. This depends mostly on the pattern type. A frequent itemset cannot be hidden using a rule hiding algorithm while an association rule can be hidden either using an itemset hiding algorithm by reducing its support or by using a rule hiding algorithm by reducing its confidence. Thus, this step attempts to understand the conditions under which a data sanitization algorithm is most useful for knowledge hiding.

Step 4: *Evaluation,* the side effects of sanitization process are measured with respect to the sensitive and non-sensitive patterns specified in the second step. For this purpose, the association rule mining with the prior minimum thresholds is applied on the sanitized database to confirm the utility and protection level of the sanitized database. When the database administrator or data owner's goals are fulfilled, the sanitized database is released to others; otherwise, the sanitization process is performed again using different parameters or using another algorithm.

4. Side effects in association rule hiding

Association rule hiding process cannot control the side effects of sensitive knowledge hiding obviously, and data utility of the sanitized data is reduced. The Hiding Failure (HF), Misses Cost (MC), and Artifactual Patterns (AP) are three side effects produced on the sanitized database. Fig. 2 illustrates the impact of data sanitization process on the released database.

The set P is the meaningful patterns in D, the sets P_S and $\neg P_S$ are the sensitive and non-sensitive patterns in D respectively. The set P' is the patterns discovered from the sanitized database (D'). An optimal sanitization algorithm should discover all patterns in $\neg P_S$ from D', this means that $P' = \neg P_S$. But the set P' may not contain all non-sensitive patterns as well as it may contain sensitive patterns or new generated patterns. In Fig. 2, the MC means that some non-sensitive patterns are hidden from the released database (lost rules/misses cost) while the HF means that some sensitive patterns are discovered from the sanitized database. In AP, some artificial patterns are generated in D' as a result of the hiding process (ghost rules/new rules) (Oliveira & Zaiane, 2006; Verykios et al., 2004b). Briefly, a sanitization algorithm has three objectives. The first is to hide all specified patterns in P_S ($HF = \emptyset$). The second is to maintain all patterns in $\neg P_S$($MC = \emptyset$), and the third is that artifactual patterns should not be produced ($AP = \emptyset$). The HF arises when data sanitization algorithm does not perform enough modifications for hiding all sensitive patterns. A sanitization process with a low minimum support threshold cannot avoid the hiding failure (Li & Chang, 2007). The MC occurs when some transactions that fully or partially support the non-sensitive patterns are modified or deleted; therefore, the support or confidence of these patterns may be reduced and cannot be extracted from the sanitized database. It is worth noting that there is a compromise between the hiding failure and the misses cost. The more sensitive patterns we hide, the more legitimate patterns we miss (Oliveira & Zaiane, 2002). The AP occurs when the confidence of a non-strong rule or support count of an infrequent itemset is increased (Oliveira & Zaiane, 2006).

In the following, the side effects of sanitization process are illustrated using an example. Table 1(a) shows a given database. Considering $\alpha_{\min} = 3$, and $\beta_{\min} = 75\%$, the frequent itemsets and the strong association rules are listed in Table 1(b) and Table 1(c), respectively.

Assume that the rule $\{a\} \Longrightarrow \{b, c\}$ is sensitive. Since its confidence is 80% ($\frac{4}{5}$) and $\beta_{\min} = 75\%$, one modification is required

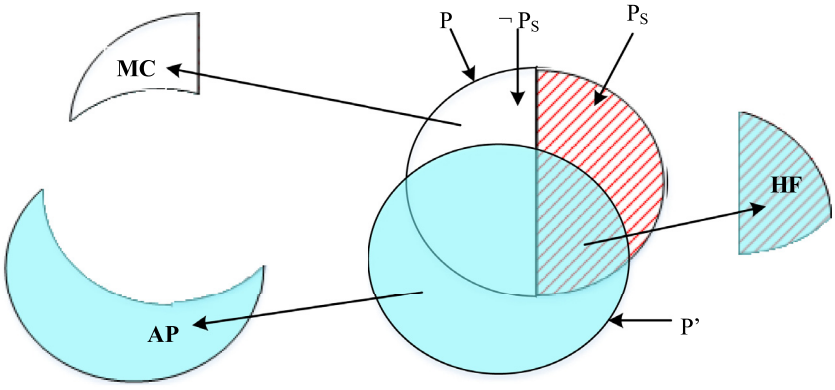


Fig. 2. Relation of the side effects produced by data sanitization algorithms.

Table 1
An example of association rule mining process.

(a) Transaction Data		(b) Frequent itemsets	(c) Strong association rules			
ID	Items	Frequent itemset/α	AR	β (%)	AR	β (%)
T1	{a, b, c, d}	{a}/5	{a}⟹{b}	100	{a, d}⟹{c}	75
T2	{b, d, e}	{b}/7	{a}⟹{c}	80	{c, d}⟹{a}	75
T3	{a, b, c}	{c}/5	{c}⟹{a}	80	{b, c}⟹{d}	75
T4	{a, b, d, e}	{d}/7	{a}⟹{d}	80	{c, d}⟹{b}	75
T5	{a, b, c, d}	{e}/4	{c}⟹{b}	80	{e}⟹{b, d}	100
T6	{b, d, e}	{a, b}/5	{b}⟹{d}	85	{b, e}⟹{d}	100
T7	{a, b, c, d, e}	{a, c}/4	{d}⟹{b}	85	{d, e}⟹{b}	100
T8	{c, d}	{a, d}/4	{e}⟹{b}	100	{a, c}⟹{b, d}	75
		{b, c}/4	{c}⟹{d}	80	{a, d}⟹{b, c}	75
		{b, d}/6	{e}⟹{d}	100	{b, c}⟹{a, d}	75
		{b, e}/4	{a}⟹{b, c}	80	{c, d}⟹{a, b}	75
		{c, d}/4	{c}⟹{a, b}	80	{a, b, c}⟹{d}	75
		{d, e}/4	{a, b}⟹{c}	80	{a, b, d}⟹{c}	75
		{a, b, c}/4	{a, c}⟹{b}	100	{b, c, d}⟹{a}	100
		{a, b, d}/4	{b, c}⟹{a}	100		
		{a, c, d}/3	{a}⟹{b, d}	80		
		{b, c, d}/3	{a, b}⟹{d}	80		
		{b, d, e}/4	{a, d}⟹{b}	100		
		{a, b, c, d}/3	{a, c}⟹{d}	75		

Table 2
Impact of different modifications for hiding {a}⟹{b, c}.

Modification	#MC (itemset)	#MC (rule)	#AP
{b}⟶*T1*	2	14	1
{b}⟶*T3*	0	3	1
{b}⟶*T5*	2	14	1
{b}⟶*T7*	2	14	0
{c}⟶*T1*	3	15	0
{c}⟶*T3*	0	3	0
{c}⟶*T5*	3	15	0
{c}⟶*T7*	3	15	0

to hide this rule, where item {*b*} or {*c*} is removed from one of supporting transactions (*T1, T3, T5* or *T7*). Hence, the confidence is decreased to 60% ($\frac{3}{5}$) after this modification. Table 2 shows the side effects caused by applying all eight modifications on Table 1(a). Removing the item {*c*} from transaction *T3* produces the fewest side effects and only three non-sensitive rules are lost. Thus, developing an algorithm that performs such modifications can decrease the number of side effects.

5. Data sanitization algorithms

The structure of a sanitization algorithm contains three parts, including hiding strategy, sanitization technique, and selection/generation method. The focus of the hiding strategy is how to hide sensitive patterns, while the focus of the sanitization technique is how to sanitize the transactions to perform the hiding strategy. The hiding strategy is obtained through the modification or deletion of existing sensitive transactions as well as through the insertion of new transactions into the database. These modifications produce side effects on the sanitized database that should be minimized. Therefore, defining a selection method that can minimize the side effects by selecting an appropriate set of transactions and items is a key issue. A sanitization algorithm accepts the original dataset, minimum thresholds, sensitive patterns, as well as the non-sensitive patterns as input. Then, it performs the sanitization steps based on the hiding strategy, sanitization technique, and selection methods defined for sanitization in order to produce the sanitized database as output.

We defined two criteria to select the published papers in the association rule hiding scope, (1) "Science Citation Index" database: Web of Science covering the highest impact journals in sciences, engineering, and humanities; therefore, we used this database to validate the journals. (2) Citation: this criterion was used for the literature that has been cited in the papers that were validated by the first criterion. We used this criterion for non-journal literature, such as books, conferences, workshops and etc., as well as for journal articles that provided data sanitization solutions but are not in the master list. From 109 literature selected for this study, forty-two papers presented the sanitization algorithms for knowledge hiding. The number of these algorithms is 54 that are used for our analysis. These algorithms are presented in Table 3 with their abbreviations. In the following, we provide

Table 3
Data sanitization algorithms for association rule hiding.

Reference	Algorithm
Saygin et al., 2001, 2002	Confidence Reduction (CR), CR2, Generating Itemset Hiding (GIH)
Dasseni et al., 2001; Verykios et al., 2004b	1.a, 1.b, 2.a
Oliveira and Zaiane, 2002	Maximum Frequency Item Algorithm (MaxFIA), Minimum Frequency Item Algorithm (MinFIA), Item Grouping Algorithm (IGA), Naïve
Oliveira and Zaiane, 2003a, 2006	Sliding Window size Algorithm (SWA)
Oliveira and Zaiane, 2003b	Random Algorithm (RA), Round Robin Algorithm (RRA)
Pontikakis et al., 2004a; Verykios et al., 2007	Priority-based Distortion Algorithm (PDA), Weight-based Sorting Distortion Algorithm (WSDA)
Verykios et al., 2004b	2.b
Pontikakis et al., 2004b; Verykios et al., 2007	Blocking Algorithm (BA)
Menon et al., 2005	Blanket, Intelligence
Wang and Jafari, 2005; Wang et al., 2007a	Increase Support of Left hand side (ISL), Decrease Support of Right-hand side (DSR)
Sun and Yu, 2005, 2007	Border-Based Approach (BBA)
Divanis and Verykios, 2006	Inline
Moustakides & Verykios, 2006, 2008	Max-Min1, Max-Min2
Amiri, 2007	Aggregate, Disaggregate, Hybrid
Li and Chang, 2007	Maximum Item Conflict First (MICF)
Wang et al., 2007b	Decrease Confidence by Decrease Support (DCDS), Decrease Confidence by Increase Support (DCIS)
Wu et al., 2007	–
Wang et al., 2008	Decrease Support and Confidence (DSC)
Menon and Sarkar, 2008	–
Divanis and Verykios, 2009a	Hybrid
Wang, 2009	Maintenance of Sanitizing Informative association rules (MSI)
Divanis and Verykios, 2009b	–
Hai and Somjit, 2012	Intersection Lattice-based Association Rule Hiding (ILARH)
Hai et al., 2012	Distance and Intersection Lattice based (DIL)
Hai et al., 2013a	Association Rule Hiding based on Intersection Lattice (ARHIL)
Hai et al., 2013b	Heuristic for Confidence and Support Reduction based on Intersection Lattice (HCSRIL)
Hong et al., 2013	Sensitive Items Frequency-Inverse Database Frequency (SIF-IDF)
Lin et al., 2014a	Compact Prelarge GA-based algorithm to Delete Transactions (cpGA2DT)
Lin et al., 2014b	–
Lin et al., 2015	Simple Genetic Algorithm to Delete Transactions (sGA2DT), Pre-large Genetic Algorithm to Delete Transactions (pGA2DT)
Lin et al., 2016	Particle Swarm Optimization-based algorithm to Delete Transactions (PSO2DT)
Cheng et al., 2014, 2016a	Evolutionary Multi-objective Optimization-base Rule Hiding (EMO-RH)
Afshari et al., 2016	Cuckoo Optimization Algorithm for Association Rules Hiding (COA4ARH)
Cheng et al., 2016b	Relevance-sorting
Telikani and Shahbahrami, 2017	Decrease Confidence of Rules (DCR)

a historical review of 54 algorithms conducted on the association rule hiding as well as present some developing pseudocodes for most popular data sanitization algorithms.

2001. Dasseni et al. (2001) presented three algorithms, namely *1.a, 1.b*, and *2.a* to hide sensitive rules. The first two algorithms reduce the confidence of a rule by increasing the support of rule antecedent and by decreasing the support of rule consequent respectively, while the third algorithm decreases the support of generating itemset of the rule. Saygin et al. (2001) proposed Confidence Reduction (CR), CR2, and Generating Itemset Hiding (GIH) algorithms that are performed similar to three previous algorithms; the difference is that the proposed algorithms replace the items with unknowns (question mark) instead of removing the items. The pseudocode of the *2.a* algorithm is depicted in Fig. 3(a). The *2.a* algorithm estimates the minimum number of transactions that need to be modified between *N_iter_conf* and *N_iter_supp*. Where *N_iter_conf* and *N_iter_supp* are the numbers of iterations to hide a sensitive rule by the confidence reduction and the support reduction strategies respectively, and the *2.a* algorithm considers the minimum number of those to minimize the sanitized transactions.

2002. Oliveira and Zaiane (2002) proposed four itemset hiding algorithms, namely Maximum Frequency Item Algorithm (MaxFIA), Minimum Frequency Item Algorithm (MinFIA), Item Grouping Algorithm (IGA), and Naïve. They considered, for the first time, the impact of transaction and item modification on the sanitized database by calculating their conflict. The pseudocode of the IGA algorithm is shown in Fig. 4.

2003. Sliding Window size Algorithm (SWA) was proposed to hide the sensitive itemsets in one scan over the whole dataset (Oliveira & Zaiane, 2003a). The sketch of the SWA is given in Fig. 3(b). The algorithm first copies the non-sensitive transactions to the sanitized database and then uses an indexing mechanism to speed up the hiding process. Unlike other algorithms, which have a unique disclosure threshold for all sensitive rules, the SWA

Algorithm: 2.a
Input: D, a set Ps of rules to hide, |D|, $\boldsymbol{\alpha_{min}}$, $\boldsymbol{\beta_{min}}$
Output: $\boldsymbol{D'}$

1. While sensitive rules not hidden {
2. Find corresponding transactions
3. Compute length of sensitive transactions
4. Sort sensitive transactions in ascending order of length
5. $N\_iter\_conf = \left\lceil |\boldsymbol{D}| * \left(\frac{\boldsymbol{supp}\,(X \Longrightarrow Y)}{\boldsymbol{\beta_{min}}} - \boldsymbol{supp}(\boldsymbol{X})\right)\right\rceil$
6. $N\_iter\_supp = \left\lceil |\boldsymbol{D}| * \left(\frac{\boldsymbol{supp}\,(X \Longrightarrow Y)}{\boldsymbol{\alpha_{min}}}\right)\right\rceil$
7. $N\_iteration = min\ (N\_iter\_conf, N\_iter\_supp)$
8. While $N\_iteration$>0{
9. Select item with minimum impact on ($|\boldsymbol{X} \Longrightarrow \boldsymbol{Y}|$-1)-itemsets
10. Remove victim item from a transaction
11. Supp (X⟹Y) = supp (X⟹Y)-1
12. Conf (X⟹Y) = supp (X⟹Y)/supp (X)
13. Remove sanitized transactions from sensitive list}
14. Remove hidden rule from sensitive rules}

Fig. 3. (a): Pseudocode of Algorithm 2.a. (b): Pseudocode of algorithm Sliding Window Size (SWA).

Algorithm: SWA
Input: D, mining permissions (M_P), Sliding window (K)
Output: D'

```
For each K transactions in D{
  For each transaction t ∈ K{
    For each sensitive itemset ∈ MP{
      If transaction t is not sensitive
        Transaction t is copied into D'
      Else{
        Transaction t is added to inverted index list
        Update support of each item in t}}
    If transaction t is sensitive{
      Sort its items in descending order of frequency
      For each sensitive itemset ∈ MP
      Select item with highest frequency as victim}}}
For each sensitive itemset ∈ MP{
  N_iteration=| sensitive transactions |*(1-α_min)
  Sort the transactions in ascending order of size}
For each sensitive itemset ∈ MP{
  While N_iteration>0
    Remove victim item from sensitive transactions}
```

Fig. 3. Continued

Algorithm: IGA
Input: D, a set Ps of itemsets to hide, α_{min}
Output: D'

```
For each transaction t in D{
  Update support of each item in t
  Sort the items in t in alphabetic order
  For each sensitive itemset{
    if t is correspond to sensitive itemset
      transaction t is added to inverted index list}}
For each sensitive itemset{
  sort sensitive transactions in descending order of conflict degree
  N_iteration=| sensitive transactions |*(1-α_min)}
Group sensitive rules in a set of groups
Assign labels to each group and select item with lower support as victim item
Order the groups in by size in term of number of sensitive itemsets in group
Compare groups pairwise and start with the largest
For each sensitive itemset{
  Sort Transactions in descending order of conflict degree
  While N_iteration>0
    Remove victim item from sensitive transactions}
```

Fig. 4. Pseudocode of algorithm Item Grouping Algorithm (IGA).

has a disclosure threshold assigned to each sensitive association rule. The set of mining permissions (M_P) is referred to the set of mappings of a sensitive association rule into its corresponding disclosure threshold. Oliveira and Zaiane (2003b) proposed two algorithms, Random Algorithm (RA) and Round Robin Algorithm (RRA), to hide the sensitive rules by reducing their generating itemsets. These algorithms consider the impact of altering transactions on the sensitive rules.

2004. Verykios et al. (2004b) extended the work of Dasseni et al. (2001) as well as proposed the *2.b* algorithm to hide the generating itemset of sensitive rules. The Priority-based Distortion Algorithm (PDA) and Weight-based Sorting Distortion Algorithm (WSDA) (Pontikakis et al., 2004a) were presented to hide the sensitive rules by formulating a heuristic in the item selection phase of PDA and in the transaction selection phase of WSDA. These algorithms were the first efforts to assign weight to transactions. Pontikakis et al. (2004b) proposed Blocking Algorithm (BA) that purposely generates the rules that do not exist in the original dataset by adding unknowns to the transactions.

2005. Frequent itemset hiding was formulated as Constraint Satisfaction Problem (CSP) by Menon et al. (2005). They proposed

Algorithm: Max-Min2
Input: D, a set Ps of itemsets to hide, α_{min}, the positive border
Output: D'

```
While |Ps|≠ ∅ {
  Sort Ps in increasing order of support
    Select itemset with lowest support
    Lsensitive ← Find corresponding transactions for sensitive itemset
    Build vi-list representation
    While α(itemset)≥ α_min {
      If max-min is not attained by a vi-list
        Determine an itemset with minimum impact as max-min itemset
      Lmax-min← Find corresponding transactions for max-min itemset
      Sanitization_list ← Compute Lsensitive - Lmax-min
      If the Sanitization_list is not empty then
        Remove victim item from transaction t in the Sanitization_list
      Else remove item from transaction t with minimum impact on max-min itemsets
        α(itemset) = α(itemset)-1}
    Remove hidden itemset from Ps}
```

Fig. 5. Pseudocode for Max-Min2 algorithm.

the *Blanket* and the *Intelligence* algorithms that solve the CSP using integer programming to minimize the number of sanitized transactions, while these algorithms use heuristics to find the victim items. Wang and Jafari (2005) incorporated unknowns to hide predictive association rules and presented the Increase Support of Left hand side (ISL) and Decrease Support of Right-hand side (DSR) algorithms. Sun and Yu (2005) proposed Border-Based Approach (BBA) inspired by the border theory of frequent itemsets (Mannila & Toivonen, 1997) to preserve the quality of the border of non-sensitive itemsets in the itemset lattice.

2006. Divanis and Verykios (2006) introduced the notion of distance between the original database and the sanitized one in the *Inline* algorithm. This algorithm relies on the process of border revision to identify the least amount of items for sanitization, instead of considering the minimum number of sanitized transactions. It solves the CSP using Binary Integer Programming (BIP). Moustakides and Verykios (2008) proposed two border-based algorithms, namely Max-Min1 and Max-Min2, which control the impact of sanitization on the itemsets that are more vulnerable to the hiding process, instead of all itemsets on the boundary. Fig. 5 is a sketch for the Max-Min2 algorithm.

2007. Amiri (2007) proposed three heuristics, namely the Aggregate, Disaggregate, and Hybrid, that outperform the SWA by offering higher data utility and lower distortion at the expense of increased computational speed. Maximum Item Conflict First (MICF) algorithm was proposed to outperform the IGA in terms of reducing the number of deleted items and overcoming the overlap between groups (Li & Chang, 2007). Wang et al. (2007a) extended the ISL and DSR algorithms (Wang & Jafari, 2005) using distortion technique. Decrease Confidence by Decrease Support (DCDS) and Decrease Confidence by Increase Support (DCIS) algorithms were proposed to automatically hide the collaborative recommendation association rules without pre-mining and selection of hidden rules (Wang et al., 2007b). Verykios et al. (2007) improved the BA algorithm (Pontikakis et al., 2004b) by applying the transaction selection heuristic used in the WSDA (Pontikakis et al., 2004a). Wu et al. (2007), presented a limited side effect methodology that classifies all valid modifications related to the sensitive rules, the non-sensitive rules, and the spurious rules that can be affected by modifications. Then, the heuristic methods are used to modify the transactions in order to increase the number of hidden sensitive rules, while reducing the number of modified entries (Divanis & Verykios, 2009a).

2008. Decrease Support and Confidence (DSC) algorithm was proposed to hide the predictive association rules (Wang et al., 2008). Menon and Sarkar (2008) extended the algorithm presented in Menon et al. (2005) to minimize both the number of sanitized transactions and the number of lost non-sensitive itemsets.

2009. Divanis and Verykios (2009a) appended a database extension to the original database instead of modifying existing transactions. The extended portion contains a set of transactions that alleviate the importance of sensitive patterns to a degree that they become uninteresting from the perspective of data mining algorithms, while minimally affecting the importance of non-sensitive itemsets. They proposed a hybrid algorithm which incorporates the CSP, BIP, and border revision to hide sensitive itemsets. Wang (2009) improved the DSC algorithm (Wang et al., 2008) and introduced the Maintenance of Sanitizing Informative association rules (MSI) algorithm to protect the sensitive information when the database is updated frequently. The newly added dataset is separately sanitized by the MSI and then is combined with the original database. Divanis and Verykios (2009b) improved the Inline approach (Divanis & Verykios, 2006) by a two-phase process. The sanitization process is terminated in the first phase, if the sensitive knowledge is concealed without producing the side effects. Otherwise, the dual counterpart of the Inline algorithm is performed in the second phase so that the hiding algorithm selectively removes the inequalities from the infeasible CSP, until the CSP becomes feasible, and then the CSP is solved to attain the sanitized dataset.

2012. Intersection lattice theory (Grätzer, 2010) was first investigated in the Intersection Lattice-based Association Rule Hiding (ILARH) algorithm (Hai & Somjit, 2012) to item selection. Distance and Intersection Lattice based (DIL) algorithm was proposed by Hai et al. (2012) that measures the impact of hiding process on the non-sensitive rules by assigning a weight to each transaction. Moreover, it considers the distance from sensitive rules to the set of maximal itemsets and the nearest non-sensitive rule to select the victim items.

2013. Hai et al. presented the Association Rule Hiding based on Intersection Lattice (ARHIL) (Hai et al., 2013a) and Heuristic for Confidence and Support Reduction based on Intersection Lattice (HCSRIL) (Hai et al., 2013b) to hide the sensitive rules. The ARHIL takes the full advantages of the ILARH (Hai & Somjit, 2012), DIL (Hai et al., 2012), and HCSRIL (Hai et al., 2013b) algorithms. It uses the characteristics of intersection lattice of frequent itemsets to select the victim items, while it identifies the transactions based on their weight inspired by the DIL algorithm (Hai et al., 2012). By adopting the concept of Term Frequency-Inverse Document Frequency (TF-IDF), Hong et al. (2013) introduced Sensitive Items Frequency-Inverse Database Frequency (SIF-IDF) algorithm to assign a weight value to each transaction.

2014. The use of Genetic Algorithms (GAs) for transaction selection in the context of itemset hiding was first suggested by Lin et al. (2014a, 2014b). Lin et al. (2014a, 2014b) Compact Prelarge GA-based algorithm to Delete Transactions (cpGA2DT) algorithm (Lin et al., 2014a) deletes the specified transactions, while the algorithm proposed in Lin et al. (2014b) generates and inserts new transactions into the database. Cheng et al. (2014) proposed Evolutionary Multi-objective Optimization-base Rule Hiding (EMO-RH) algorithm. The architecture of this algorithm is based on the PISA platform (Bleuler et al., 2003). In the variation part of platform, problem-specific encoding scheme and efficient variation operators are devised. The selector part of PISA is implemented using the NSGA II algorithm (Deb et al., 2002).

2015. Lin et al. (2015) introduced two itemset hiding algorithms, namely Simple Genetic Algorithm to Delete Transactions (sGA2DT) and Pre-large Genetic Algorithm to Delete Transactions (pGA2DT) that use GAs to select transactions, and then delete transactions from the original database.

2016. The drawback of GA-based algorithms is that several parameters should be specified by the user, irrespective of finding appropriate values (rates) for parameters, such as chromosome size, mutation rate, and crossover rate. Besides, these algorithms require to manually specify the number of transactions for deletion. To deal with these problems, Particle Swarm Optimization-based algorithm to Delete Transactions (PSO2DT) algorithm (Lin et al., 2016) was presented that can determine the maximum number of transactions that can be deleted, as well as fewer parameters need to be set. Afshari et al. (2016) proposed the Cuckoo Optimization Algorithm for Association Rules Hiding (COA4ARH) for hiding the sensitive association rules by applying Cuckoo Algorithm (COA) (Yang & Deb, 2009). They defined a pre-processing operation with two phases in the beginning of the proposed algorithm. This operation remarkably reduces both the number of iterations and access time to the optimal solution. The Relevance-sorting algorithm was proposed by Cheng et al. (2016b) that formulates a heuristic for determining transactions for sanitization. In order to reduce the distortion ratio, algorithm computes the minimum number of transactions that need to be modified to conceal a sensitive rule.

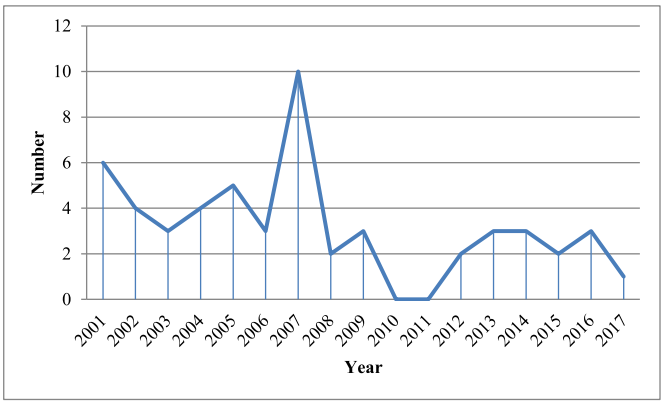


Fig. 6. Number of data sanitization algorithms from 2001 to 2017.

2017. The Decrease the Confidence of Rule (DCR) algorithm was proposed in Telikani and Shahbahrami (2017) that improves the MaxMin solution (Moustakides & Verykios, 2006, 2008) using two heuristics to hide the association rules. In this algorithm, the combination of MaxMin and heuristic approaches is formulated to select victim items, while the sensitive transactions are chosen using a heuristic solution.

Fig. 6 depicts the number of sanitization algorithms between 2001 and 2017. As shown, the highest number of algorithms has been proposed in 2007 (10 out of 54, 19%); also, no algorithm has been presented during 2010 and 2011. From that time on, association rule hiding has attracted more interests in recent years (2012–2017).

Fig. 7 shows the research key points for data sanitization process. As can be seen from the figure, the blocking and distortion techniques were used in 2001 to modify the sensitive transactions. In 2005, the focus of algorithms was to maintain the utility and accuracy of the sanitized databases so that the border theory and CSP were formulated for these purposes, respectively. As a result of applying these techniques, the exact and border approaches have appeared in 2005. Concurrent with abolishing the blocking technique in 2007, transaction deletion technique was introduced by Amiri (2007). In fact, the studies published in 2007 have focused on the sanitization techniques while they had concentrated on the selection methods in 2005. Transaction insertion technique was used in 2009 to degrade the importance of sensitive itemsets. In 2012, the intersection lattice theory increased the research motivations after two years research stagnation between 2010 and 2011. The GAs-based framework was first applied to select transactions in 2014, and so the evolutionary approach was introduced.

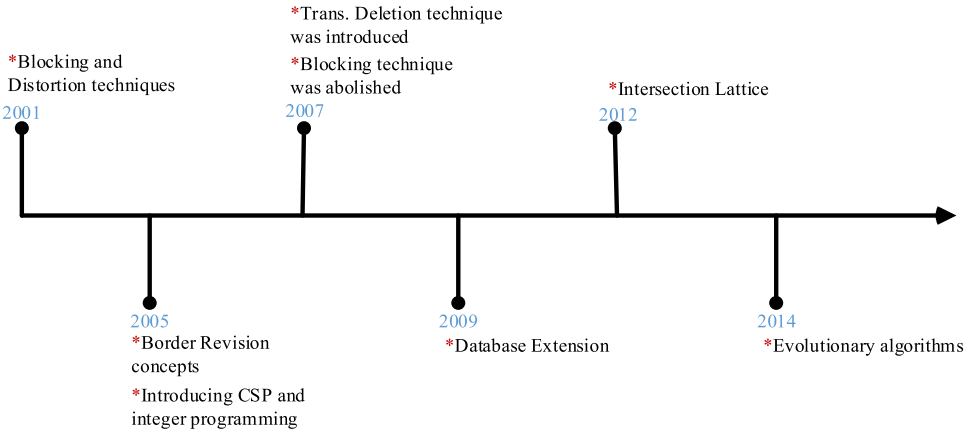


Fig. 7. Key points in lifetime of data sanitization process.

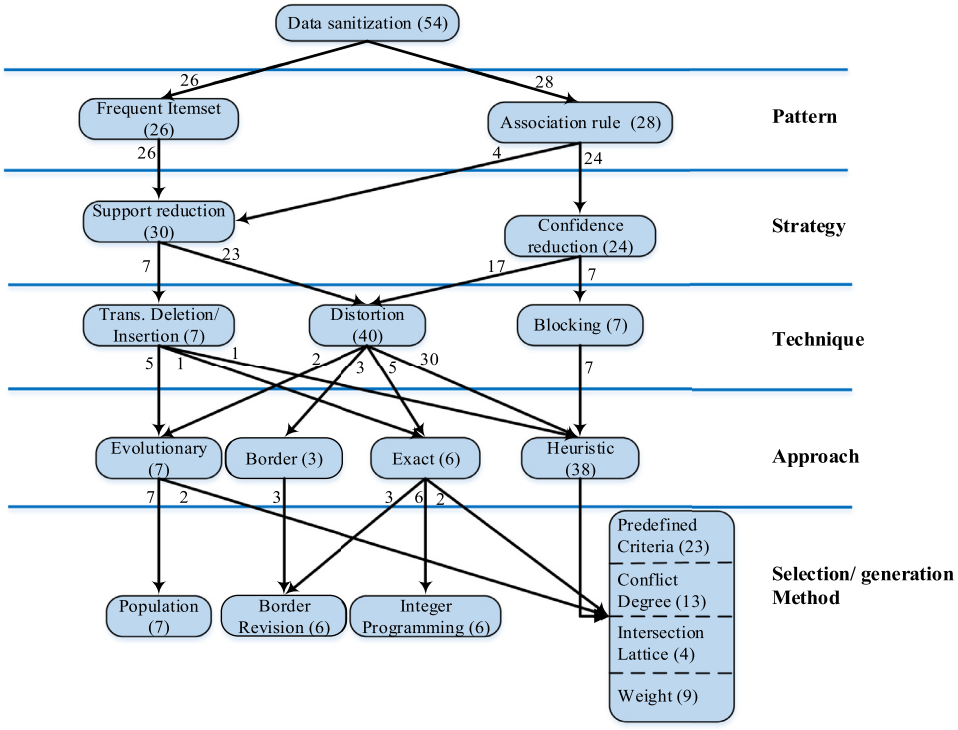


Fig. 8. A taxonomy of data sanitization algorithms according to four concepts of strategy, technique, approach, and method.

6. State of the art in data sanitization

We study and analyze all 54 algorithms and present a classification of state-of-the-art directions based on four aspects of data sanitization process, including strategy, technique, approach, and selection method. Fig. 8 shows a taxonomy of this analysis which presents the distribution of algorithms related to each category. The total distributions at the last level are different from other levels since some algorithms are hybrid and incorporate different selection methods. For example, three exact-based algorithms use the border revision and integer programming methods to select best modifications. The characteristics of each aspect, as well as a discussion of solutions for speeding up the sanitization process, are given in the following sections.

6.1. Hiding strategies

From the hiding strategy perspective, data sanitization algorithms are analyzed in two groups, (1) frequent itemset hiding, and (2) rule hiding. The first category hides the sensitive itemsets by reducing their support below the α_{min}. The second category applies either the support or confidence reduction strategy to hide the sensitive rules. In the support reduction strategy for association rules, the support of generating itemset of the rule is reduced. In confidence reduction strategy, the support of either the consequent of the rule is reduced or the antecedent of the rule is increased. Fig. 9 shows taxonomy of mentioned strategies for sanitization process. The next sections describe different ways of reducing the confidence and support with different side effects produced by each strategy.

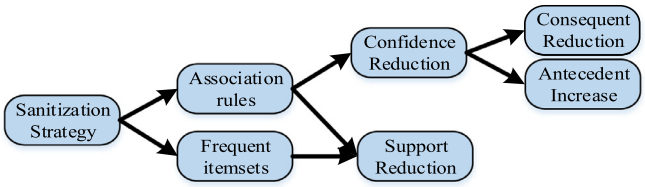


Fig. 9. The strategies in the data sanitization process.

Table 4
Impact of adding {a} to transaction *T2*.

Modification	AP (itemset)	AP (rule)
{a}⟶*T2*	{a, e} {a, b, e} {a, d, e}	{e}⟹{a}, {e}⟹{a, b}, {a, e}⟹{b}, {b, e}⟹{a}, {e}⟹{a, d}, {a, e}⟹{d}, {d, e}⟹{a}, {b}⟹{a}, {d}⟹{a}, {b, d}⟹{a}

6.1.1. Confidence reduction

As shown in Fig. 9, two strategies can be considered to decrease the confidence of a sensitive rule $X \Longrightarrow Y$, including consequent reduction and antecedent increase. In the consequent reduction, the support count of itemset Y is decreased by removing items from sensitive transactions. This strategy produces only the MC and AP side effects, which were shown by using an example in Table 2. In the antecedent increase strategy, the support count of X is increased by adding items to non-sensitive transactions that partially support X and fully support Y. This strategy not only introduces the MC and AP side effects similar to the consequent reduction strategy but also may fail to hide all sensitive patterns. Besides, it sanitizes more transactions than the antecedent increase strategy when hiding a specific rule since it decreases both the numerator and the denominator of confidence Eq. (2) while the consequent reduction strategy only reduces the numerator. Three following examples show the side effects produced by the antecedent increase strategy. All examples were performed on the transaction database presented in Table 1(a).

Example 1. Hiding failure. Assuming that {d}⟹{b} is a sensitive rule, two transactions are required to be sanitized. The item {d} cannot be added to any transaction because all non-sensitive transactions for item {d} support the item {b} too, therefore, the rule {d}⟹{b} cannot be hidden. This usually occurs when the support count of Y is very high and the number of transactions required to hide the rule is not enough, it means that $\delta < (m\text{-}\,\alpha(Y))$ where δ is number of iterations, which is defined as the number of transactions required to reduce the confidence of the rule below the β_{min}. The δ value for antecedent increase strategy is computed using Eq. (3).

$$\delta(X \Rightarrow Y) = \lceil |D| * (\alpha\ (X \Rightarrow Y)/\ \beta_{min} - \alpha(X)) \rceil \tag{3}$$

Example 2. Artifactual patterns. Assuming that the rule {a}⟹{b, d} is sensitive, one modification is needed to hide the rule, where item {a} is added either to transaction *T2* or to transaction *T6*. For example, if we add the item {a} to transaction *T2*; this modification introduces three new itemsets and 10 new rules, as shown in Table 4.

Example 3. Misses cost. We consider Example 2 to demonstrate the non-sensitive rules that are lost from the sanitized data. When the item {a} is added to *T2*, the rules {a}⟹{c}, {a}⟹{b, c}, {a, b}⟹{c}, {a, d}⟹{c}, {a, d}⟹{b, c}, and {a, b, d}⟹{c} are hidden. This is due to this fact that the support count of antecedent of these rules is increased, and thus the confidence of these rules is decreased.

The *1.b* (Dasseni et al., 2001; Verykios et al., 2004b), DCDS (Wang et al., 2007b), DSR (Wang et al., 2007a, 2007b), ILARH (Hai & Somjit, 2012), DIL (Hai et al., 2012), ARHIL (Hai et al., 2013a), HCSRIL (Hai et al., 2013b), PDA, WSDA (Pontikakis et al., 2004a; Verykios et al., 2007), CR (Saygin et al., 2001, 2002), COA4ARH (Afshari et al., 2016), and DCR (Telikani & Shahbahrami, 2017) use the consequent reduction strategy to reduce the confidence of sensitive rules. The 1.a (Dasseni et al., 2001; Verykios et al., 2004b), DCIS (Wang et al., 2007b), ISL (Wang et al., 2007a, 2007b), BA (Pontikakis et al., 2004b), and CR (Saygin et al., 2001, 2002) algorithms apply the antecedent increase strategy to hide sensitive association rules. Unlike other association rule hiding algorithms that hide a set of specified rules, the ISL, DSR (Wang et al., 2007a, 2007b), DSC (Wang et al., 2008), and MSI (Wang, 2009) hide rules containing a set of sensitive items on the left-hand side of rules. On the other hand, the DCIS and DCDS (Wang et al., 2007b) hide the association rules that the sensitive items belong to the right-hand side of the rules.

A combination of two above strategies was presented in Verykios et al. (2007) and Wu et al. (2007) to reduce the confidence of sensitive rules both by decreasing the support count of Y and by increasing the support count of X. The main idea behind this strategy is to balance the side effects in terms of lost rules and ghost rules produced in the sanitized database; because reducing just the support count of the rule consequent introduces many lost rules, and also increasing just the support count of the rule antecedent produces many new rules. The extended BA algorithm (Verykios et al., 2007) and algorithm proposed in Wu et al. (2007) utilize this strategy to hide the association rules.

6.1.2. Support reduction

The support reduction strategy does not generate any artificial itemset while it may produce new rules as the side effects since the strength of the weak rules may be increased by reducing the support of their left-hand side. On the other hand, this strategy hides some non-sensitive itemsets as misses cost. Furthermore, the number of non-sensitive rules hidden by the support reduction strategy is more than that of the confidence reduction strategy since all association rules derived from the hidden itemsets are concealed after sanitization process.

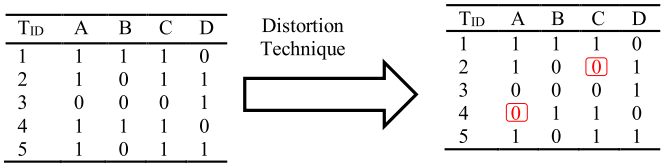
The RA, RRA (Oliveira & Zaiane, 2003b), 2.b (Verykios et al., 2004b), and GIH (Saygin et al., 2001, 2002) hide sensitive rules by reducing their generating itemset. The DSC (Wang et al., 2008) and MSI (Wang, 2009) decrease the support count of rule antecedent in order to reduce the confidence of these rules. Divanis and Verykios (2009a) proposed an itemset hiding approach that inserts the synthetic transactions in a way that the support count of sensitive itemsets is reduced and support count of non-sensitive ones is maintained as possible.

6.2. Sanitization techniques

In sanitization process, existing transactions are modified or deleted from the database, or new transactions are added to the database. The *distortion* and *blocking* techniques modify the existing transactions by removing/inserting items from/into the transactions. This study considers the transaction deletion and insertion techniques as a unified concept called the *transaction deletion/insertion* technique so as to cover a wider range of algorithms. This section discusses the sanitization process in terms of different sanitization techniques.

6.2.1. Distortion technique

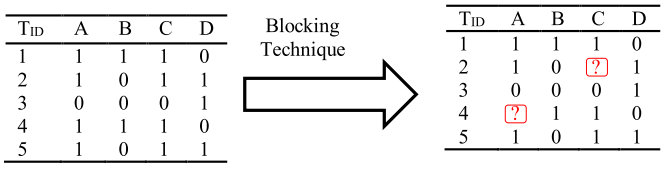
This technique was first proposed by Atallah et al. (1999) to delete specific items from the transactions. How to implement the distortion technique in the binary and categorical databases



T_{ID}	A	B	C	D
1	1	1	1	0
2	1	0	1	1
3	0	0	0	1
4	1	1	1	0
5	1	0	1	1

T_{ID}	A	B	C	D
1	1	1	1	0
2	1	0	0	1
3	0	0	0	1
4	0	1	1	0
5	1	0	1	1

Fig. 10. An example of distortion technique.



T_{ID}	A	B	C	D
1	1	1	1	0
2	1	0	1	1
3	0	0	0	1
4	1	1	1	0
5	1	0	1	1

T_{ID}	A	B	C	D
1	1	1	1	0
2	1	0	?	1
3	0	0	0	1
4	?	1	1	0
5	1	0	1	1

Fig. 11. Unknown values in blocking technique.

is different. In the binary database, the algorithms modify the transactions via inserting (replacing 0's by 1's) or deleting (replacing 1's by 0's), while in the categorical database, the victim items are removed from sensitive transactions or items are added to non-sensitive transactions. It has been proven that the distortion technique for association rule hiding is a NP-hard problem (Atallah et al., 1999). In the database sanitized by distortion technique, the data recipient cannot be certain about the change of any specific item of the database because any item could have been inserted or deleted by sanitization process. An example of distortion technique in the binary database is shown in Fig. 10. As can be seen in this figure, some 1's have been replaced by 0's.

6.2.2. Blocking technique

The main idea of the blocking technique was first proposed in Chang and Moskowitz (1998) which the victim items are replaced with unknown values (Saygin et al., 2001). Pontikakis et al. (2004b) suggested that it is required to maximize the number of unknowns to prevent an adversary from recovering the sensitive rules. Fig. 11 shows an example of blocking technique for association rule hiding.

The blocking technique makes a distinction between distorted items and unaffected items of the original database since only the distorted items are replaced by unknowns. This is a nice property in critical life applications, where the distinction between "false" and "unknown" can be vital because it does not add any false rules to the original database. On the negative side, it may fuzzify the support and confidence of an association rule due to the incorporation of unknowns (Cheng et al., 2016b). Indeed, it is difficult to mine the significant non-sensitive association rules from the database sanitized by the blocking technique since the occurrence of an item is expressed by '1' and association rule mining algorithms count this value for each itemset, and any value except this value means zero. Also, since the sanitized transactions are explicit for data recipient, an adversary can observe that the received data has been sanitized because of the existence of sensitive knowledge. Therefore, the disclosure risk is increased, especially in the binary dataset, where each transaction contains binary values (either 0 or 1). The CR, CR2, GIH (Saygin et al., 2001, 2002), BA (Pontikakis et al., 2004b; Verykios et al., 2007), and an extended version of the ISL and DSR (Wang & Jafari, 2005) algorithms replace the binary items by the unknown values.

6.2.3. Transaction deletion/insertion

The transaction deletion/insertion technique has received significant attention in recent years. It aims to delete the transactions from the database or to insert the new transactions into the original databases. The problem of deleting existing transactions from the dataset was first introduced in the *Aggregate* algorithm (Amiri, 2007). Divanis and Verykios (2009a) proposed an exact border-based approach so that unimportant transactions are appended to the original database. Most of the evolutionary-based algorithms use transaction deletion/insertion technique. Lin et al. (2014b) proposed a GA-based algorithm that inserts transactions into the database. The cpGA2DT (Lin et al., 2014a), sGA2DT, pGA2DT (Lin et al., 2015), and PSO2DT (Lin et al., 2016) remove transactions from the original database to reduce the support of sensitive itemsets. The transaction deletion/insertion technique changes the number of transactions of the database; therefore, it is difficult to mine patterns with prior minimum thresholds.

6.3. Sanitization approaches

The main focus of sanitization algorithms is on finding an optimal solution to hide all sensitive knowledge with minimum side effects. For this purpose, four approaches have been presented, including heuristic, border, exact, and evolutionary. These approaches are described in the following sections.

6.3.1. Heuristic approach

This approach includes efficient and fast algorithms that select a set of transactions by using predefined criteria. Although the heuristic approach has witnessed a lot of attention from researchers in recent years, it is not necessarily globally best and does not guarantee the optimality of the hiding solution, however, it usually finds a solution near the best one in a faster response time. Because of the fact that the heuristic-based algorithms always aim at taking locally best decisions with respect to the hiding of sensitive knowledge, they produce more undesirable side effects than other algorithms especially in itemsets hiding (Divanis & Verykios, 2010). Some of the most interesting heuristic-based algorithms for hiding the sensitive knowledge have been presented in Divanis and Verykios (2010), Verykios (2013), and Verykios and Divanis (2008).

6.3.2. Border approach

Borders (Mannila & Toivonen, 1997) capture those itemsets of frequent itemset lattice that control the position of the borderline separating the frequent itemsets from their infrequent counterparts (Verykios & Divanis, 2008). Data sanitization affects the borders of the sanitized database; therefore, this approach considers the association rule hiding through the modification of the borders in the lattice of the frequent and infrequent itemsets of the original database. It was first introduced by Sun and Yu (2007) for hiding the frequent itemsets while maintaining the non-sensitive itemsets with low support. The border-based algorithms sanitize the transactions with minimum impact on the results of the released database (Moustakides & Verykios, 2008). The BBA (Sun & Yu, 2005, 2007), Max-Min1, and Max-Min2 (Moustakides & Verykios, 2006, 2008) algorithms use the border theory to hide the frequent itemsets.

6.3.3. Exact approach

Exact approach tries to conceal the sensitive patterns by causing minimum distortion to the sanitized database. It considers the problem of frequent itemset hiding as a CSP, and formulates the CSP as an integer program to minimize the number of sanitized transactions or items (Menon et al., 2005). Due to the use of integer programming solver to solve the optimization problem, the exact-based algorithms are very complex (Divanis & Verykios, 2010). The border approach is often considered as a complementary to the exact hiding approach because most of the exact-based algorithms use the border theory. In this method, first, the border theory is applied to compute a small portion of

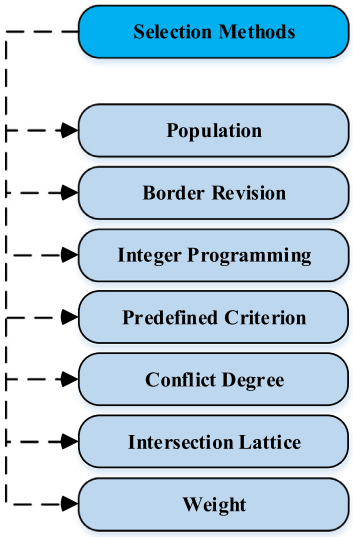


Fig. 12. The selection/generation methods for sanitization process.

itemsets which can play a crucial role in maintaining the quality of the hiding solution, and then the inequalities are generated by using exact methodologies to control the status of the selected itemsets of the border (Verykios, 2013). There is no absolute exact-based algorithm since the heuristic or border solutions are often incorporated to find a sanitization solution. Blanket, Intelligence (Menon et al., 2005), Inline (Divanis & Verykios, 2006), and the algorithm presented in Divanis and Verykios (2009a, 2009b) and Menon and Sarkar (2008) are exact-based.

6.3.4. Evolutionary approach

The paradigm of evolutionary algorithms consists of stochastic search algorithms inspired by the process of neoDarwinian evolution. These algorithms work with a population of individuals. Each individual is a candidate solution to a given problem that is evolved towards better and better solutions to that problem. The quality of the candidate solution is measured by a fitness function predefined by the user. This is a very generic search paradigm and can be used to solve many different kinds of problems (Maimon & Rokach, 2010). The cpGA2DT (Lin et al., 2014a), the algorithm proposed in Lin et al. (2014b), sGA2DT, pGA2DT (Lin et al., 2015), PSO2DT (Lin et al., 2016), EMO-RH (Cheng et al., 2014, 2016a), and COA4ARH (Afshari et al., 2016) algorithms use the evolutionary algorithms to knowledge hiding.

6.4. Selection/generation methods

Finding appropriate victim items and transactions for sanitization as well as generating new transactions for adding to the database is the most important step in the sanitization process. It plays an important role in reducing the side effects. Fig. 12 depicts a classification of methods formulated to select/generate items and transactions.

6.4.1. Population

In NP-hard problems, the population-based approaches are widely used to find near best solutions in order to optimize the problems by evaluating all solutions. These approaches facilitate the search for good solutions by applying the principles of natural evolution. The Genetic algorithms (GAs) (Holland, 1992) are the most fundamental population-based approach (Lin et al., 2016). In GAs, the idea is to encode each solution as a chromosome. The initial population is randomly generated, and then each solution is reproduced using various operations such as selection, mutation, and crossover. Finally, the goodness of chromosomes is evaluated using a designed fitness function. This process is repeated until the stop condition is satisfied (Lin et al., 2016). In the field of association rule hiding, each chromosome encodes a solution consisting of a set of sensitive transactions, and the fitness function is designed based on all three side effects of hiding process, including lost rules, ghost rules, and hiding failure. Lin et al. (2014a) employed GAs to select a set of transactions for deletion. They presented the cpGA2DT (Lin et al., 2014a), sGA2DT, and pGA2DT (Lin et al., 2015) algorithms. Lin et al. (2014b) also used the GAs to generate the appropriate transactions to be inserted into the database.

The PSO (Kennedy & Eberhart, 1995) has been inspired by the behavior of birds flocking to find the better food sources. In a PSO, particles represent the problem solutions, where each particle has a velocity representing a flying direction toward other solutions. The PSO procedure first randomly initializes the particles, and then an iterative evolution process is performed. During each iteration, each particle is updated using its personal best value (*pbest*) and global best value (*gbest*) based on the designed fitness function to update old particles and to generate offsprings of the population. The *gbest* value is the best solution among all *pbest* values in the population. The particles and their corresponding velocities are evaluated and updated using these two best values (Lin et al., 2016). The implementation of PSO is easier than the GAs for discovering optimal solutions because the PSO, unlike GAs, has not crossover and mutation operations (Kuo et al., 2011). Similar to GAs, it uses randomized evolutionary approach for the sanitization problem. The discrete PSO was applied in the PSO2DT algorithm (Lin et al., 2016) to find a set of transactions. In this algorithm, the particles and their velocities are assigned to the set of transaction identifiers.

Multi-objective optimization problems are usually solved using population-based evolutionary algorithms (Bandaru et al., 2016). The evolutionary multi-objective optimization was utilized in the EMO-RH algorithm (Cheng et al., 2014, 2016a) by adopting the binary encoding scheme to hide the sensitive rules. In this algorithm, the side effects are formulated as optimization objectives in order to find a suitable subset of transactions for modification. Each bit in the chromosome corresponds only to a supporting transaction, thus reducing the size of search space. Each chromosome is divided into k segments, where k is the number of sensitive rules. The length of the jth segment is the number of transactions which support the jth sensitive rule. In the EMO-RH, the uniform crossover and the independent bit mutation are utilized in the evolution process.

Inspired by the cuckoo bird, the Cuckoo Algorithm (COA) was first developed by Yang and Deb (2009). Like other evolutionary algorithms, the COA also begins its work from a random initial population which is called "*habitat*" and it is formed by *cuckoos*. This algorithm was adopted in the COA4ARH algorithm (Afshari et al., 2016) to select the sensitive transactions based on three fitness functions defined for three side effects. The COA4ARH also introduces an immigration function to escape from any local optimum. Each solution of initial population is shown with a sequence of 0's and 1's. In this algorithm, the first solution of initial population is a sequence of sensitive transactions selected by a pre-processing step, and other solutions are generated by randomly quantifying those sensitive transactions that had been addressed in pre-processing step while other transactions are left unchanged from the first solution.

6.4.2. Border revision

In order to minimize the impact of sanitization process on the non-sensitive patterns, the effect of modifications on these patterns should be controlled. Actually, this method maintains the

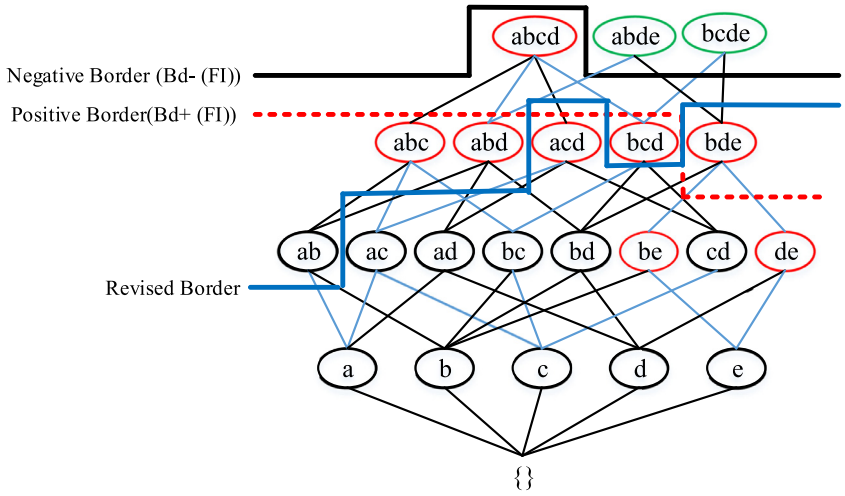


Fig. 13. The itemsets lattice for frequent itemsets with original border and revised border for Table 1(b).

Table 5
Non-sensitive frequent itemsets which should be extracted from the sanitized database.

Expected Frequent Itemsets in D'
{a, c, d}, {b, d, e}
{a, c},{a, d},{b, c},{b, d},{b, e}, {c, d},{d, e}
{a}, {b}, {c}, {d}, {e}

aggregated quality of the result database during the knowledge hiding process (Sun & Yu, 2007). According to the border theory (Mannila & Toivonen, 1997), the elements on the border of itemsets lattice are the boundary to the infrequent itemsets. In the following, the border revision method is clearly described using the frequent itemsets of Table 1(b). Assume that the itemsets {a, b} and {b, c, d} are sensitive. Table 5 presents the expected frequent itemsets, i.e., the itemsets that should be frequent in the sanitized database (D'). According to the Apriori property (Agrawal & Srikant, 1994), when a sensitive frequent itemset is hidden, its supersets are also hidden from the sanitized database as well. Therefore, itemsets {a, b, c}, {a, b, d}, and {a, b, c, d} are hidden from D' after hiding the itemsets {a, b} and {b, c, d}.

Fig. 13 shows the itemset lattice for Table 1(b) with its negative border, positive border, and revised border. The negative border (or original border) of Frequent Itemset (FI), denoted by $Bd^-(FI)$, is the set of all infrequent itemsets from D in which all proper subsets appear in FI. The positive border of FI, denoted by $Bd^+(FI)$, is the set of all maximally frequent itemsets appearing in FI. The revised border is the ideal borderline after itemsets hiding, which allows us to hide the sensitive itemsets without changing any expected frequent itemsets to infrequent. If all itemsets on the revised border remain frequent, then the sanitization solution is optimal (Sun & Yu, 2007). Therefore, the method focuses on preserving the quality of the revised border by greedily selecting the modifications with the minimal side effect.

The BBA algorithm (Sun & Yu, 2005, 2007) selects the relevant modifications by evaluating the impact of any modification on the border during the itemset hiding process, so that the items with minimum impact on the revised border are deleted. The Max-Min approach (Moustakides & Verykios, 2006, 2008) uses the border theory for item selection. In this approach, for each item of sensitive itemset, the non-sensitive itemsets containing item are specified. From among the specified itemsets, one itemset with minimum support is selected as a candidate itemset to be protected from hiding. Therefore, the number of candidate itemsets is equal to the number of items of sensitive itemsets. Then, from among the candidate itemsets, one itemset with maximum support is chosen as *maxmin* itemset. Finally, the item belonging to the maxmin itemset is selected as victim item. Based on this approach, the Max-Min1 and Max-Min2 algorithms were proposed to perform the modifications in such a way that the support count of the maxmin itemset, if possible, not to be modified (Moustakides & Verykios, 2008). When there are more than one maxmin itemsets, two algorithms are performed in two different ways to select the victim item. In this case, the Max-Min1 randomly selects one of them as maxmin itemset, while the MaxMin2 selects the itemset with minimum effect. Telikani and Shahbahrami (2017) developed the Max-Min approach to select victim items in the context of rule hiding so that their algorithm, DCR, controls the impact of sanitization on the association rules with low confidence.

The application of border revision method was first introduced in Divanis and Verykios (2006) to maintain the data accuracy of the sanitized data, i.e., the number of actual item modifications is minimized. This method has also been applied by Divanis and Verykios (2009a) to generate new transactions to be added to the original database as well as by Divanis and Verykios (2009b) to find an exact solution for hiding the sensitive frequent itemsets without side effects.

6.4.3. Integer programming

A CSP (Russell & Norvig, 2003) is defined by a set of variables, a finite and discrete domain for each variable, and a set of constraints, where each variable has a non-empty domain of potential values. The constraints involve a subset of the variables to specify the allowable combinations of values that these variables can attain. An assignment that does not violate the set of constraints is called “consistent”. The goal is to satisfy all constraints in order to maximize or minimize an objective function subject to a number of constraints (Divanis & Verykios, 2009a; Kumar, 1992). The integer programming technique can be applied to find an exact solution by using linear or nonlinear programming (Luenberger, 1973). Since all variables in the sanitization problem are binary, the BIP technique (Gueret et al., 2002) can be useful in transforming the CSP to an optimization problem.

The CSP tries to close the distance of the original database and its sanitized version, in other words, the goal is to maintain data accuracy. In a basic definition of accuracy, the accuracy of a relation is the proportion of accurate tuples in the relation, where a tuple is said to be accurate if and only if every attribute value in the tuple is accurate (Reddy & Wang, 1995).

Menon et al. (2005) defined the accuracy as the number of transactions that are not sanitized. They believe that the data accuracy and data utility of the sanitized database can be maximized by minimizing the number of sanitized transactions. In this regard, Menon et al. (2005), Menon and Sarkar (2008). and Divanis and Verykios (2006, 2009a, 2009b) considered the hiding process as a CSP problem to find the optimal sanitization solution.

6.4.4. Predefined criteria

In this category, the victim items are selected based on *support* and the transactions are selected based on *length*. There are two criteria for support-based selection: minimum support and maximum support. The main reason for the former is that the item with low support belongs to the less number of patterns; therefore, modifying this item causes the least impact on the non-sensitive patterns. The rationale behind the latter is that the non-sensitive patterns containing the item with the highest frequency have high support, and so these patterns are minimally affected by sanitization process. The MinFIA (Oliveira & Zaiane, 2002) and DCR (Telikani & Shahbahrami, 2017) algorithms aim at reducing the impact of each modification by selecting items with minimum support. The Naïve (Oliveira & Zaiane, 2002) and Blanket (Menon et al., 2005) algorithms also follow this strategy in such a way that all items of sensitive transactions are removed except for the item with the highest frequency. The MaxFIA (Oliveira & Zaiane, 2002), SWA (Oliveira & Zaiane, 2003a), 2.b (Verykios et al., 2004b), GIH (Saygin et al., 2001, 2002), EMO-RH (Cheng et al., 2014; Cheng et al., 2016a), and Relevance-sorting (Cheng et al., 2016b) select item with the highest frequency as victim item.

There is only one criterion to select the transactions where the transactions with the shortest length are selected for sanitization. The assumption is that these transactions produce fewer frequent patterns; as a result, fewer association rules are generated in comparison with transactions with the longest length. For example, the transaction {c, d} has the shortest length in Table 1(a), three frequent itemsets and one association rule are produced from this transaction. On the other hand, the transaction {a, b, c, d, e} has the highest length and all patterns are generated from this transaction. The HCSRIL (Hai et al., 2013b), 1.b, 2.a (Dasseni et al., 2001; Verykios et al., 2004b), 2.b (Verykios et al., 2004b), DSC (Wang et al., 2008), DCIS, DCDS (Wang et al., 2007b), ISL, DSR (Wang & Jafari, 2005; Wang et al., 2007a), CR, GIH (Saygin et al., 2001, 2002), and SWA (Oliveira & Zaiane, 2003a) algorithms choose the transactions with the shortest length. The 1.a (Dasseni et al., 2001; Verykios et al., 2004b) and CR2 (Saygin et al., 2001, 2002) algorithms select the non-sensitive transactions that support the maximum number of items on the left-hand side of the sensitive rule.

6.4.5. Conflict degree

The conflict degree indicates the number of patterns which are affected if an item or transaction is sanitized. It can measure the impact of sanitization on the non-sensitive patterns or on the sensitive patterns. The first measurement directly controls the impact of data sanitization on the non-sensitive patterns by selecting modifications with minimum impact. The second measurement has two objectives: the first is to hide more sensitive patterns at the same time, and thus the modifications with the maximum impact are selected. The second objective is to sanitize the transactions/items with minimum impact. The main assumption behind this objective is that the item or transaction which depends on fewer sensitive patterns also depends on fewer non-sensitive patterns, and so little side effects can be produced. Considering the conflict degree on the non-sensitive patterns, sanitization process is performed with a larger number of iterations than when sensitive patterns are considered since the number of non-sensitive patterns is usually much more than the number of sensitive patterns. In the case of association rules, the degree of conflict can be considered on the right-hand side, left-hand side, or both hand side. The hand side of the rules is related to the hiding strategy, for example, when the strategy is the consequent reduction, the conflict degree is considered on the right-hand side of association rules.

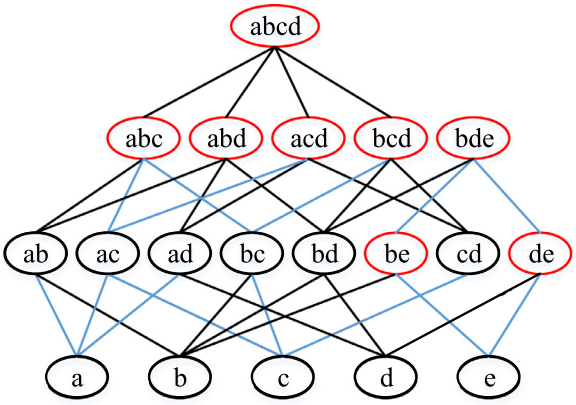


Fig. 14. The graph of intersection lattice of frequent itemsets for Table 1(b).

In the transaction selection phase, the MinFIA, MaxFIA, naïve, and RRA algorithms (Oliveira & Zaiane, 2003b) select transactions with the lowest conflict degree on the sensitive patterns, while the MICF (Li & Chang, 2007), IGA (Oliveira & Zaiane, 2002), and RA (Oliveira & Zaiane, 2003b) algorithms select the transactions with the highest conflict degree on the sensitive patterns. In the item selection phase, the MICF (Li & Chang, 2007), Intelligence (Menon et al., 2005), IGA (Oliveira & Zaiane, 2002), and SIF-IDF (Hong et al., 2013) algorithms select items with the maximum conflict degree on the sensitive patterns. In the IGA, the victim item in one rule is fixed and is removed from all sensitive transactions associated with the sensitive itemset. Although this increases the efficiency of sanitization process, it may maximally affect the non-sensitive itemsets related to the victim item. Unlike the above-mentioned algorithms that consider the conflict degree on the sensitive patterns, the *Disaggregate* algorithm (Amiri, 2007) considers the degree of conflict of the item on the non-sensitive itemsets and selects the item with minimum conflict degree as the victim. The *1.b* algorithm (Dasseni et al., 2001; Verykios et al., 2004b) computes the impact of items on the $(|Y|-1)$-itemsets. The *2.a* algorithm (Dasseni et al., 2001; Verykios et al., 2004b) measures the impact of items on the $(|X \cup Y|-1)$-itemsets. Both *1.b* and *2.a* algorithms select an item with the minimum impact. The COA4ARH (Afshari et al., 2016) selects a sensitive item with the most frequency from the right-hand side of sensitive rules and the least frequency in the non-sensitive rules.

6.4.6. Intersection lattice

The lattice theory (Grätzer, 2010) was first adopted by Hai et al. (2012) to select the victim items. They analyzed the characteristics of the intersection lattice of frequent itemsets to minimize the side effects on the frequent itemsets with the low support. The basic concepts of the lattice theory-based selection are presented as follows. First, the intersection lattice of all frequent itemsets is generated and then the number of supersets of each itemset in U is calculated by function $d(Z)$, where Z is an itemset and U is the set of all frequent itemsets (Hai et al., 2013b). An itemset W is a superset for the itemset Z when $Z \subseteq W$. Fig. 14 shows the graph of intersection lattice of the frequent itemsets in Table 1(b).

By the Apriori property, if Z and $W \in U$, then $Z \cap W \in U$. It can be inferred that U is an intersection lattice. The Generating Set (GS)

Table 6
The frequent itemsets and *GS* (*U*).

U	a	b	c	d	e	ab	ac	ad	bc	bd	be	cd	de	abc	abd	acd	bcd	bde	abcd
s	3	4	3	3	2	2	2	2	2	3	1	2	1	1	1	1	1	0	0

of *U*, denoted by *GS* (*U*), is the smallest set of itemsets of *U* such that every itemset of *U* can be generated by taking an intersection of some itemsets in *GS* (*U*). The set *GS* (*U*) can be computed by

$$GS(U) = \{Z \in U | d(Z) \leq 1\}, \text{ where } d(Z) = |Z \in U | Z \subset W| \quad (4)$$

The number of supersets, denoted by *s*, computed by function *d (Z)* and the set *U* are presented in Table 6. *Coatom* (*U*) is all maximum itemset of *U*, in other words, itemsets with $s = 0$ is *GS* (*U*), therefore,

$$GS(U) = \{\text{be, de, abc, abd, acd, bcd, bde, abcd}\},$$
$$\text{and } Coatom(U) = \max(GS(U)) = \{\text{bde, abcd}\}$$

The itemsets contained in *GS* (*U*) have the lowest support in *U*. Therefore, these itemsets are vulnerable to the support reduction of any item. Furthermore, if every itemset of *GS* (*U*) is frequent, then all itemsets of *U* are also frequent. Inspired by the idea of intersection lattice, some algorithms have been proposed to maintain *GS* (*U*) during the hiding process in order to restrict the lost rules. The ILARH (Hai & Somjit, 2012), HCSRIL (Hai et al., 2013b), ARHIL (Hai et al., 2013a), and DIL (Hai et al., 2012) specify the victim items based on the characteristics of the intersection lattice of frequent itemsets.

6.4.7. Weight

The weight-based prioritization is a heuristic to select the transactions and items, where the weight of each sensitive transaction or item is computed. In WSDA algorithm (Pontikakis et al., 2004a; Verykios et al., 2007), first, a weight is assigned to each sensitive rule according to how close to the minimum confidence threshold. Then, the priority value of each sensitive transaction is computed based on the weights and finally, the sensitive transactions with the lowest priority are sanitized. Verykios et al. (2007) adopted this selection method in the item selection phase of the BA algorithm. The DIL (Hai et al., 2012) and ARHIL (Hai et al., 2013a) algorithms assign a weight to each sensitive transaction relying on its degree of safety, the number of sensitive rules, and the number of non-sensitive association rules contained in that transaction. Unlike the above weight-based algorithms, the DIL sanitizes the transactions with the highest weight. The PDA (Pontikakis et al., 2004a) selects an item with the minimum impact on the sensitive association rules by assigning the priority to each item.

Hong et al. (2013) improved the concept of TF-IDF (Salton et al., 1983) used in text mining to estimate the degree of transactions associated with the sensitive itemsets. In this method, the degree of correlation between each sensitive transaction and sensitive itemsets is computed. The SIF-IDF algorithm (Hong et al., 2013) prioritizes the supporting transactions with the highest weight for modification. The major drawback of this algorithm is that it only employs the information on the sensitive itemsets contained in a transaction, while the non-sensitive ones are not considered although these are more relevant to side effects (Cheng et al., 2016a). Cheng et al. (2016b) formulated a heuristic to compute the degree of relevance of each transaction by considering the conflict of transaction on the generating itemsets of non-sensitive rules. In this method, the transactions with the highest relevance value are selected for sanitization. Indeed, the transactions that have minimum relation with the non-sensitive itemsets are sanitized.

Table 7
The inverted file of Table 1(a).

Items	Frequency		Transaction IDs
A	5	→	T1, T3, T4, T5, T7
B	7	→	T1, T2, T3, T4, T5, T6, T7
C	5	→	T1, T3, T5, T7, T8
D	7	→	T1, T2, T4, T5, T6, T7, T8
E	4	→	T2, T4, T6, T7

6.5. Speed-up of sanitization process

The alternative objective of data sanitization algorithms is to reduce the computation time required for hiding process. There are two ways to speed up the sanitization process; in the first method, an efficient solution is defined for the transaction and item selection phases. Most of the heuristic algorithms use this manner because they assign a value to the transactions based on different heuristics such as length, conflict degree, or weight. For this reason, first, they sort transactions in ascending/descending order of the value of the transactions in *O* (*S*), where *S* is the number of sensitive transactions, and then the transactions are selected from the top of the list one by one. Thus, the speed of selection of each transaction is increased from *O (S)* to *O (1)*. In the second method, an independent technique is applied to reduce the database scanning for identifying the sensitive transactions. This technique either can collaborate with the sanitization process or can be embedded into the sanitization process. In the following, various techniques proposed for the second way are discussed.

Oliveira and Zaiane (2002, 2003b) introduced a transaction retrieval engine relying on an inverted file for retrieving the transaction IDs from the database. The transaction database is indexed into an inverted file in which, for each item of the database, there is a corresponding list of transaction IDs related to the item. The transaction IDs is sorted in ascending order of the transaction IDs. Thus, in the worst case, a transaction ID is found by using binary search with an access time of *O* (*logN*), where *N* is the number of transaction IDs in the list. The inverted file's vocabulary is composed of all different items in the transaction database which is implemented based on a perfect hash table (Dietzfelbinger et al., 1994). Table 7 shows the inverted file for Table 1(a). The RRA, RA (Oliveira & Zaiane, 2003b), MaxFIA, MinFIA, Naïve, and IGA (Oliveira & Zaiane, 2002) algorithms use this technique.

Pattern-Inversion tree (PI-tree) data structure (Wang, 2009; Wang et al., 2008) is an extension of the pattern tree technique (Huang et al., 2002) to reduce the number of database scanning. Each node in PI-tree contains three fields: item name, the number of transactions containing the items on the path from the root to the current node, and the list of transaction IDs that contains all items on the path from the root to the current node. The transactions are firstly read from the database one by one. Each transaction is sorted according to the item name and is inserted into the PI-tree. The frequency list is updated accordingly. Then, the frequency list is sorted according to the support count of the items. Finally, the PI-tree is restructured similar to the first step. Fig. 15 presents the PI-tree for transactions in Table 1(a). The DSC (Wang et al., 2008) and MSI (Wang, 2009) use the PI-tree to hide the sensitive knowledge with one scan of the database.

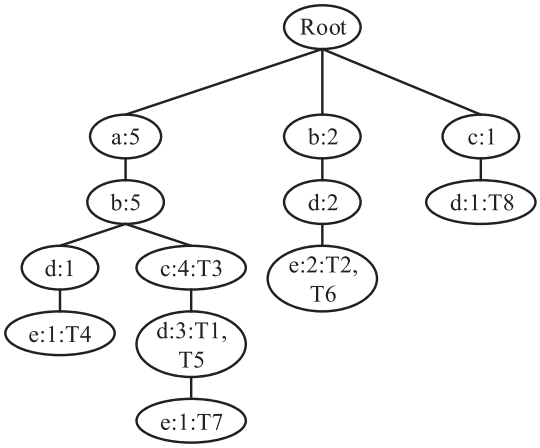


Fig. 15. Pattern-inversion tree for dataset in Table 1(a).

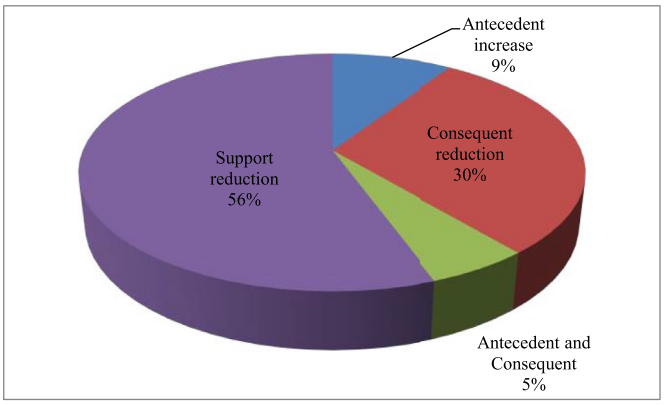


Fig. 16. Distribution of association rule hiding algorithms based on hiding strategy.

In order to speed up the evolution process in evolutionary-based algorithms, the pGA2DT (Lin et al., 2015) and PSO2DT (Lin et al., 2016) adopt the pre-large concept (Hong et al., 2001) to reduce the execution time for rescanning the original database in the evaluation process. This optimization consists of maintaining a buffer of pre-large itemsets during the evolution process to avoid performing multiple database scans. Lin et al. (2014b) adopted the downward closure property (Liu et al., 2005) and the pre-large concept to reduce the cost of rescanning database. The compact GA approach (Harik et al., 1999) was applied in cpGA2DT (Lin et al., 2014a) to generate only two individuals per population for competition in order to reduce the memory usage.

The MICF (Li & Chang, 2007) initially loads all sensitive transactions into the main memory. Therefore, the transactions are sanitized in the main memory instead of the disk. MICF applies an index lookup table to efficiently sort the sensitive transactions and to reduce the memory space requirement. Each transaction is associated with a pair of two values in the lookup table: the first is the index value pointing to the transaction and the second is the conflict degree value of the transaction. When the size of sensitive transactions exceeds the current available memory space, the MICF suffers from a lot of page swaps between the disk and main memory. In this case, a partitioning technique is combined with the initial MICF to handle very large databases. During the sanitization process, only the sensitive transactions of the current partition are loaded into the main memory. Then, the initial MICF is applied to sanitize the partition. If the support count of sensitive itemsets did not decrease below the privacy threshold, the algorithm loads the sensitive transactions of the next partition.

The trie-tree data structure (Bodon, 2005) was adopted by Cheng et al. (2014, 2016a) to store the support count of frequent itemsets in order to increase the access and update speed of the support count of these itemsets. A two-dimensional array is first used to store the support of the rules with two items and then the trie-tree is applied to reserve and retrieve the corresponding generating itemsets of the rules with more than two items. When the array is used, data can be directly accessed in a mapping way, and so there is no need to retrieve the whole rules set to find the one for updating its support.

Wu et al. (2007) used two techniques in order to speed up the sanitization process, (1) the original database is represented in the form of bit-vectors where each distinct item is encoded as a unique prime number. (2) The transaction-rule index is constructed using the concept of the inverted lists (Kowalski & Maybury, 2006) to correlate the tables for efficient retrieval.

The extended BA algorithm (Verykios et al., 2007) uses some data structures to access the rules in the database. It produces four tables, including one inverted index table and three hash tables. (1) The inverted index table is generated for every item of the database such that the merge sort algorithm is used to find the support of large itemsets. (2) For each rule, rule along with the number of its supporting transactions that are above the confidence threshold are stored. (3) The large itemsets and their support value are stored to be recovered quickly, and (4) the non-sensitive rules with low confidence are stored along with how many transactions have below minimum thresholds.

7. Discussion and analysis

Four concepts related to the data sanitization, including hiding strategy, sanitization technique, sanitization approach, and selection/generation method are used by a set of association rule hiding algorithms to hide the sensitive knowledge. This section presents a statistical analysis of applying these directions in all 54 data sanitization algorithms as well as compares the advantages and disadvantages of each direction. The characteristics of these directions were described extensively and independently of each other in Sections 6.1–6.4.

7.1. Analysis by hiding strategies

Table 8 summarizes the characteristics of sanitization strategies discussed in Section 6.1. The support reduction strategy reduces the quality of association rule mining results in terms of losing the non-sensitive patterns and producing new rules, no new itemsets. In rule hiding problem, the consequent reduction strategy sanitizes fewer transactions in comparison with the antecedent increase strategy; therefore, its data accuracy is high. As a disadvantage of the antecedent increase strategy, it may not always hide all sensitive rules. In general, the antecedent increase strategy is worse than the consequent reduction strategy. Applying the consequent reduction and antecedent increase strategies together can lead to fewer lost rules and new rules than when two strategies are applied separately.

Fig. 16 presents the distribution of algorithms by the hiding strategy. From 54 data sanitization algorithms, most of them, 56% (30 out of 54), use the support reduction strategy to hide the sensitive patterns. From these 30 algorithms, seven algorithms are rule hiding-based and 23 algorithms are itemset hiding-based. On the other hand, 24 (44%) algorithms apply the confidence reduction strategy to hide sensitive rules. Most of these algorithms reduce the confidence by decreasing the support count of the rule consequent, 16 out of 24, followed by the antecedent increase

Table 8
Comparison of hiding strategy for association rule hiding.

Method	Advantages	Disadvantages
Support reduction	It does not create any artifactual itemset.	The data utility in mining of the association rules from sanitized database is reduced.
Consequent reduction	There is no hiding failure. The data accuracy is more than the antecedent increase strategy.	–
Antecedent increase	–	It may fail in hiding all sensitive rules. The sanitization ratio is high.
Consequent reduction with antecedent increase	Number of lost rules and new rules is low.	–

Table 9
Comparison of techniques for association rule hiding.

Method	Advantages	Disadvantages
Distortion	The disclosure risk of sensitive patterns is low.	–
Blocking	It does not add any false information to the data.	It may fuzzify the support and confidence of the association rule. It cannot guarantee the protection of sensitive patterns.
Transaction deletion/insertion	It does not need to select the victim items	It changes the number of transactions.

Table 10
Distribution of algorithms in terms of sanitization technique and publication year.

Sanitization technique	2001	2002	2003	2004	2005	2006	2007	2008	2009	2012	2013	2014	2015	2016	2017	**Total**
Distortion	3	4	3	3	3	3	8	2	2	2	3	1	0	2	1	**40**
Blocking	3	0	0	1	2	0	1	0	0	0	0	0	0	0	0	**7**
Transaction deletion/insertion	0	0	0	0	0	0	1	0	1	0	0	2	2	1	0	**7**
Total	**6**	**4**	**3**	**4**	**5**	**3**	**10**	**2**	**3**	**2**	**3**	**3**	**2**	**3**	**1**	**54**

strategy, 5 out of 24, and the combination of the consequent reduction and the antecedent increase strategies, 3 out of 24.

7.2. Analysis by sanitization techniques

Table 9 presents a comparison between different sanitization techniques. In the blocking technique, an adversary can disclose the hidden rules by identifying those generating itemsets that contain question marks, and then he/she can replace the question marks with the actual items. On the other hand, since the distortion technique inverts the items of binary database or deletes/inserts the items from/into the categorical database, disclosure risk of sensitive patterns is decreased. An important drawback of the transaction deletion/insertion technique is that it changes the number of transactions of the database so that the importance of meaningful and useless patterns may be affected.

Table 10 exhibits the distribution of sanitization algorithms based on sanitization technique and their publishing year from 2001 to 2017. It can be seen from this table that distortion is the most commonly used sanitization technique, 74% (40 out of 54), while the blocking technique has not been used to modify the database in recent years (from 2007 to 2017). On the other hand, the transaction deletion/insertion technique has recently been applied to sanitize the databases, especially from 2014 to 2016.

7.3. Analysis by sanitization approaches

Table 11 presents the benefits and drawbacks of the sanitization approaches. Since the border and exact approaches conceal a sensitive rule by hiding its generating itemset, they do not achieve good results when hiding a set of association rules in comparison with the heuristic and evolutionary approaches. The main difference between the evolutionary approach and other approaches is that the evolutionary-based algorithms formulate fitness function by considering all three side effects of sanitization process to find the best transactions for sanitization, while other algorithms consider the misses cost to select a transaction or item. Also, the evolutionary approach may not always satisfy the main condition of sanitization process. Indeed, these algorithms may fail in hiding all sensitive patterns but they reduce the number of lost rules and ghost rules. On the other hand, the main goal of other three approaches is to hide all sensitive patterns, and reduction of MC and AP are considered as the next objectives. The border approach tries to maintain the data utility while the exact approach focuses on maintaining the data accuracy. Although the heuristic approach cannot find an optimal solution to maintain the utility of the sanitized database, it usually can efficiently perform the sanitization process.

Fig. 17 shows the distribution of algorithms by the sanitization approach and the type of pattern. This analysis showed that the vast majority of algorithms, 70% (38 out of 54), are heuristic-based, followed by the evolutionary-based, exact-based, and border-based included 7, 6, and 3 out of 54, respectively. However, this classification cannot be absolute since most of the non-heuristic algorithms are hybrid and often use a heuristic or border theory for selection. For example, the Blanket, Intelligence (Menon et al., 2005), Inline (Divanis & Verykios, 2006), the algorithm proposed in Menon and Sarkar (2008), and *Hybrid* algorithm (Divanis & Verykios, 2009a), that are classified in the exact category, integrate other solutions in the selection phases. Two first algorithms use the heuristic to select the victim item and other three algorithms apply the border theory to select the transactions. Two evolutionary-based algorithms, the EMO-RH (Cheng et al., 2014, 2016a) and COA4ARH (Afshari et al., 2016) use the heuristics to select the victim items. Nonetheless, we followed the approach-based classification pre-

Table 11
Comparison of sanitization approaches for association rule hiding.

Method	Advantages	Disadvantages
Heuristic	Efficiency and stability are high. Result quality in hiding the association rules is more than frequent itemset hiding.	Data quality is less than other approaches. It does not guarantee the optimal solution.
Border	The impact of hiding process on the result quality is reduced.	–
Exact	Data accuracy is high.	Computational complexity is very high. It only focuses on data accuracy.
Evolutionary	All three side effects are considered in definition of fitness function.	It may fail in hiding all sensitive patterns.

Table 12
Comparison of selection methods for association rule hiding.

Method	Advantages	Disadvantages
Population	–	It fails in hiding some patterns. It can only be applied in transaction selection step.
Border revision	The result quality is high.	–
Integer programming	The data accuracy is high	The computational time is high. The size of search space is high.
Predefined criterion	The computational time is low.	It selects based on hypothesis.
Conflict degree	It calculates the impact of each item/transaction on patterns.	–
Intersection lattice	–	–
Weight-based	–	–

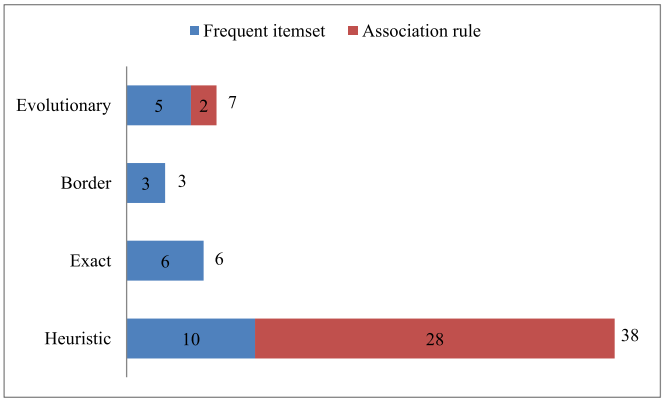


Fig. 17. Distribution of hiding algorithms based on the sanitization approaches and type of pattern.

Table 13
Distribution of selection methods used in sanitization process based on transaction and item selection.

Method	Transaction selection	Item selection
Population	5	0
Border revision	4	3
Integer programming	6	4
Predefined criterion	17	11
Conflict degree	11	9
Intersection lattice	0	4
Weight-based	6	1
Total	**49**	**32**

sented in other surveys (Verykios & Divanis, 2008; Divanis & Verykios, 2010; Verykios, 2013). As shown in Fig. 17, most of the heuristic-based algorithms have been designed to hide the association rules, 28 out of 38. All exact-based and border-based algorithms hide the frequent itemsets.

7.4. Analysis by selection/generation methods

Table 12 summarizes the advantages and drawbacks of the selection/generation methods. Since the predefined methods perform the selection phase through some assumptions, it requires less computation time than other methods. The border revision and integer programming are used in the border and exact approaches respectively, thus these methods take the characteristics of them, including high complexity, maximum data utility, and high data accuracy.

Table 13 shows the distribution of the selection/generation methods in the sanitization algorithms by the frequency of the use in the selection phases. From 54 algorithms, selection methods were used in 49 algorithms to select the transactions for sanitization while they have been applied in 32 algorithms to select the victim items. This shows that the selection of transactions has received more attention than the victim item selection since the modification of transaction affects more impact on the utility of the sanitized database than the modification of victim item. The predefined criteria are the most widely used heuristic in the sanitization algorithms, in the transaction selection step of 17 algorithms and in the victim item selection step of 11 algorithms. Followed by the conflict degree, that was used in the transaction selection step of 11 algorithms and in the victim item selection step of 9 sanitization algorithms. The population-based methods have never been used in the victim item selection phase because most of the evolutionary-based algorithms use the transaction deletion/insertion technique, therefore, they do not need to select the victim items. The intersection lattice-based method has been only used in victim item selection phase.

Our analysis showed that selection of item with the highest frequency and also selection of transaction with the shortest length are the most used predefined-based heuristics, 8 out of 11 and 16 out of 17 algorithms, respectively. This can be due to the fact that these transactions/items produce fewer patterns, thus the released database can be less affected by the sanitization process. Therefore, these heuristics achieve a good trade-off between efficiency and utility.

Table 14 shows the number of the use of conflict degree-based heuristics for selection. It is clear that in both item and transaction selection steps, the degree of conflict on the sensitive patterns is the most widely used method, followed by the calculation of conflict degree on the both maximum sensitive and minimum non-sensitive patterns. On the other hand, the conflict degree on the non-sensitive patterns has been only used by one sanitization algorithm as a criterion for selection.

Table 14
Distribution of the Conflict Degree (CD)-based methods in sanitization algorithms.

Step	Method	Number
Transaction selection	CD on sensitive patterns	7
	CD on non-sensitive patterns	0
	CD on maximum sensitive and minimum non-sensitive	4
	CD on all patterns	0
	Total	**11**
Item selection	CD on sensitive patterns	4
	CD on non-sensitive patterns	1
	CD on maximum sensitive and minimum non-sensitive	3
	CD on all patterns	2
	Total	**10**

8. Evaluation benchmarks

In the association rule hiding, it is important to assess the side effect and database effect produced by the sanitization process. There is thus the need of identifying a set of measures for this purpose. The side effects are measured in terms of the hiding failures, new rules, and lost rules. The database effect is measured at two levels: transaction level and item level. At the transaction level, the percentage of the altered transactions is measured, and at the item level, the percentage of frequencies of the changed items is measured (Amiri, 2007; Wang et al., 2008). These assessments can be performed by database administrators to determine whether the hiding process can satisfy their goals, or by researchers to estimate the impact of the new designed algorithm by using some standard transaction datasets in comparison with other algorithms. The measures and transaction datasets are discussed in the following sections.

8.1. Measures

The Misses Cost (MC), Hiding Failure (HF), and Artifactual Pattern (AP) are measured using Eqs. (5), (6), and (7), respectively. The MC is measured in terms of the percentage of legitimate patterns that are not discovered from the sanitized database (D'). The HF is measured in terms of the percentage of sensitive rules that are discovered from D'. The AP is measured as the percentage of rules that are not present in the original database but are discovered from the sanitized database.

$$\mathrm{MC} = \frac{\#\neg P_S(D) - \#\neg P_S(D')}{\#\neg P_S(D)} \tag{5}$$

$$\mathrm{HF} = \frac{\#P_S(D')}{\#P_S(D)} \tag{6}$$

$$\mathrm{AP} = \frac{|P| - |P \cap P'|}{|P'|} \tag{7}$$

Where $\#\neg P_S(D)$ and $\#\neg P_S(D')$ are the number of the existing non-sensitive patterns in the original and sanitized databases respectively, $\#P_S(D')$ is the number of sensitive patterns discovered from the sanitized database, and $\#P_S(D)$ is the number of the existing sensitive patterns in the original database.

The second goal of the sanitization process is to maintain the data accuracy in order to minimize the number of changes in the sanitized database. If data accuracy is degraded significantly, the sanitized database is useless for purpose of the knowledge extraction. At the transaction level, accuracy can be measured in terms of the number of transactions that are not sanitized (Menon et al., 2005). In order to evaluate data accuracy at the item level, the dissimilarity between the original and the sanitized dataset is often used. Dissimilarity is measured as the percentage of frequencies of the items of the two datasets before and after the sanitization, as expressed in below equation

$$\mathrm{Diss}(D, D') = \frac{\sum_{i=1}^{n} |f_D(i) - f_{D'}(i)|}{\sum_{i=1}^{n} f_D(i)} \tag{8}$$

Where n is the number of items in the dataset, $f_D(i)$ is the frequency of item i in the original dataset, and $f_{D'}(i)$ is the frequency of item i in the sanitized dataset. Bertino et al. (2005) proposed an evaluation framework for measuring the effectiveness of association rule hiding algorithms and also other kinds of PPDM tasks. The measures have also been extensively studied in the literature (Menon & Sarkar, 2008; Verykios & Divanis, 2008).

8.2. Datasets

Many real datasets are considered to evaluate the effectiveness of data sanitization algorithms. These datasets were collected in the first workshop on Frequent Itemset Mining Implementations (FIMI) and are available through the FIMI repository.[1] They have been described in more detail in Goethals and Zaki (2003) and Menon et al. (2005). These datasets demonstrate varying characteristics in terms of the number of transactions, the number of items, and average transaction length (Telikani & Shahbahrami, 2017). These characteristics are depicted in the third, fourth, and fifth columns of Table 15, respectively. Among the described datasets, the bms-pos, Bmsl, Bms2 (Zheng et al., 1999), Kosarak (Bodon, 2003), and Retail (Brijs, 2003) databases are sparse, and the Accidents (Geurts et al., 2003), Connect, Chess, and Mushroom (Bayardo, 1998) datasets are denser. The density of a dataset, the average transaction length divided by the number of items, affects the effectiveness of association rule hiding algorithms. As indicated in Menon et al. (2005), very dense datasets are not representative of transactional data in reality. The sparse datasets are usually observed more commonly in the real-world scenario. Amiri (2007) proved that the density of the database seems not to have a significant impact on data accuracy of the sanitized database, especially at the item level. The sixth column of Table 15 presents the density of real datasets. The Chess dataset has the highest density (0.493), followed by the Connect, Mushroom, and Accidents datasets with 0.33, 0.19, and 0.08, respectively. Other datasets have very lower density, for example, the Kosarak dataset is the sparsest dataset (0.0002).

The synthetic dataset is another type of dataset which is generated by IBM's Synthetic Data Generator.[2] The seventh column of Table 15 presents the number of times that a dataset has been used to evaluate the sanitization algorithms. From 42 articles, 21 articles used the synthetic datasets for performance evaluation. This is due to the researchers can produce a dataset with the number of transactions and items arbitrarily. On the contrary, the

[1] http://fimi.cs.helsinki.fi
[2] http://www.almaden.ibm.com/cs/quest/syndata.html

Table 15
Characteristics of the real databases.

Num.	Name	#Transactions	#Items	Avg. trans. Length	Density	#Usage
1	bms-pos	515,597	1657	7.50	0.0045 (Sparse)	1
2	Bms1	59,602	497	2.5	0.005 (Sparse)	15
3	Bms2	77,512	3340	5.6	0.0016 (Sparse)	13
4	Kosarak	990,002	41,217	8.10	0.0002 (Sparse)	2
5	Retail	88,162	16,470	10.30	0.0006 (Sparse)	8
6	Accidents	340,183	468	33.80	0.08 (Dense)	1
7	Connect	67,557	129	43.00	0.33 (Dense)	1
8	Chess	3196	75	37.00	0.493 (Dense)	6
9	Mushroom	8124	119	23.00	0.193 (Dense)	14
10	IBM Synthetic Data	–	–	–	–	21

bms-pos, Connect, and Accidents datasets are the least common in evaluations.

9. Conclusions and future developments

The optimality of sanitization algorithms in minimizing the undesirable effects of hiding process is an essential problem. The evidence collected for this paper indicated that different factors influence the optimality. In this study, we presented a new analytical review of 54 algorithms proposed for association rule hiding focusing on the investigation of major contributing concepts in sanitization process. The results represented in this paper have several significant implications:

- With regard to the approach, the heuristic and evolutionary approaches are up-to-date, while the border and exact approaches were only applied between 2005 and 2009. The heuristic and evolutionary approaches can be applied both in rule hiding and in itemset hiding areas. The border and exact approaches perform only the itemset hiding process, the former focuses on reducing the impact of sanitization on the non-sensitive itemsets in order to maintain the utility of the sanitized database. The latter aims to decrease the number of transactions or items sanitized by hiding process.
- Our research demonstrated that the distortion technique has attracted the greatest attention, since the emergence of association rule hiding problem. On the other hand, the blocking technique since 2007 is outdated, while the transaction deletion/insertion technique has appeared in that year.
- The main focus of all algorithms is to formulate a solution for selection of appropriate transactions and victim items in order to optimize the hiding process. This is an important motivation for researchers to design a new algorithm. Most of the algorithms use the heuristic-based methods so that 74% of algorithms use these methods. The application of population-based methods has attracted considerable interest in recent years, especially in the field of itemset hiding.
- Despite differences in selection methods there is a major consensus on the use of sanitization technique and hiding strategy so that 74% of algorithms use the distortion technique and 65% of rule hiding algorithms apply the consequent reduction strategy to reduce the confidence.
- Our analysis showed that the misses cost and artifactual patterns depend mostly on the selection methods and then depend on the hiding strategy, while the hiding failure depends only on the hiding strategy when the confidence is reduced by the antecedent increase strategy. A sanitization process concealing itemsets by item deletion does not generate any artificial itemset while it may produce artificial rules.

There are several promising directions for further research in association rule hiding. One of which is the extension of the border revision, exact, and evolutionary solutions to cover the rule hiding problem, instead of hiding the generative itemsets of association rules. Since the aim of evolutionary-based algorithms is to select the sensitive transactions for deletion, they can be combined with an optimal item selection solution in order to distort the transactions instead of deletion. Some other population-based approaches such as artificial bee colony, bacterial foraging, and ant colony optimization can be used to select transactions for sanitization. Reducing the computational time of the exact approach, especially for very large databases, is another issue that can be solved by using parallel processing so that the constraint satisfaction problem is decomposed into different components and each of them is solved independently. As a good opportunity for researchers, there is a lack of studies evaluating the effectiveness of hiding algorithms, which help database administrators in deciding on selecting an algorithm for knowledge hiding.

References

Afshari, M. H., Dehkordi, M. N., & Akbari, M. (2016). Association rule hiding using cuckoo optimization algorithm. *Expert Systems with Applications, 64*, 340–351.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 20th international conference on very large databases* (pp. 487–499).

Agrawal, R., & Srikant, R. (2000). Privacy-preserving data mining. In *Proceedings of the ACM SIGMOD international conference on management of data* (pp. 439–450).

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD conference on management of data* (pp. 207–216).

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

Atallah, M., Bertino, E., Elmagarmid, A., Ibrahim, M., & Verykios, V. S. (1999). Disclosure limitation of sensitive rules. In *Proceedings of the IEEE knowledge and data engineering exchange workshop* (pp. 45–52).

Bandaru, S., Ng, A. H., & Deb, K. (2016). Data mining methods for knowledge discovery in multi-objective optimization: Part A-Survey. *Expert Systems with Applications, 70*, 139–159.

Bayardo, R. (1998). Efficiently mining long patterns from databases. In *Proceedings of the ACM SIGMOD international conference on management of data*.

Bertino, E., Fovino, I. N., & Povenza, L. P. (2005). A framework for evaluating privacy preserving data mining algorithms. *Data Mining and Knowledge Discovery, 11*(2), 121–154.

Bleuler, S., Laumanns, M., Thiele, L., & Zitzler, E. (2003). PISA- a platform and programming language independent interface for search algorithms. In *Proceedings of the international conference on evolutionary multi-criterion optimization* (pp. 494–508).

Bodon, F. (2003). A fast APRIORI implementation. In *Proceedings of the workshop on frequent itemset mining implementations*.

Bodon, F. (2005). A trie-based APRIORI implementation for mining frequent item sequences. In *Proceedings of the 1st international workshop on open source data mining: Frequent pattern mining implementations* (pp. 56–65). ACM.

Brijs, T. (2003). Retail market basket data set. In *Proceedings of the workshop on frequent itemset mining implementations*.

Cao, J., Karras, P., Raïssi, C., & Tan, K. (2010). *P*-uncertainty: inference-proof transaction anonymization. In *Proceedings of the very large data base Endowment* (pp. 1033–1044).

Chang, L., & Moskowitz, I. (1998). Parsimonious downgrading and decision trees applied to the inference problem. In *Proceedings of the new security paradigms workshop* (pp. 82–89).

Cheng, P., Lee, I., Lin, C. W., & Pan, J. S. (2016a). Association rule hiding based on evolutionary multi-objective optimization. *Intelligent Data Analysis, 20*(3), 495–514.

Cheng, P., Pan, J. S., & Lin, C. W. (2014). Privacy preserving association rule mining using binary encoded NSGA-II. In *Proceedings of the 18th pacific-asia conference on knowledge discovery and data mining* (pp. 87–99).

Cheng, P., Roddick, J. F., Chu, S. C., & Lin, C. W. (2016b). Privacy preservation through a greedy, distortion-based rule-hiding method. *Applied Intelligence, 44*(2), 295–306.

Clifton, C., & Marks, D. (1996). Security and privacy implications of data mining. In *Proceedings of the ACM workshop on data mining and knowledge discovery* (pp. 15–19).

Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., & Zhu, M. Y. (2002). Tools for privacy preserving distributed data mining. *ACM SIGKDD Explorations Newsletter, 4*(2), 28–34.

Dasseni, E., Verykios, V. S., Elmagarmid, A. K., & Bertino, E. (2001). Hiding association rules by using confidence and support. In *Proceedings of the 4th international workshop on information hiding* (pp. 369–383).

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Dietzfelbinger, M., Karlin, A. R., Mehlhorn, K., auf der Heide, F. M., Rohnert, H., & Tarjan, R. E. (1994). Dynamic perfect hashing: Upper and lower bounds. *SIAM Journal on Computing, 23*(4), 738–761.

Divanis, A. G., & Verykios, V. (2006). An integer programming approach for frequent itemset hiding. In *Proceedings of the 15th ACM conference on information and knowledge management* (pp. 748–757).

Divanis, A. G., & Verykios, V. (2009a). Exact knowledge hiding through database extension. *IEEE Transactions on Knowledge and Data Engineering, 21*(5), 699–713.

Divanis, A. G., & Verykios, V. (2009b). Hiding sensitive knowledge without side effects. *Knowledge and Information Systems, 20*(3), 263–299.

Divanis, A. G., & Verykios, V. S. (2010). *Association rule hiding for data mining*. Springer Science & Business Media.

Evfimievski, A., Srikant, R., Agrawal, R., & Gehrke, J. (2004). Privacy preserving mining of association rules. *Information Systems, 29*(4), 343–364.

Fung, B. C. M., Wang, K., Chen, R., & Yu, P. S. (2010). Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys, 42*(4), 141–153.

Geurts, K., Wets, G., Brijs, T., & Vanhoof, K. (2003). Profiling high frequency accident locations using association rules. *Journal of the Transportation Research Board*, 123–130.

Goethals, B., & Zaki, M. (2003). Advances in frequent itemset mining implementations: Report on FIMI'03. In *Proceedings of the workshop frequent itemset mining implementations* (pp. 109–117).

Grätzer, G. (2010). *Lattice theory: Foundation*. Springer.

Gueret, C., Prins, C., & Sevaux, M. (2002). *Applications of optimization with Xpress-MP*. Dash Optimization.

Hai, L. Q., & Somjit, A. (2012). A conceptual framework for privacy preserving of association rule mining in e-commerce. In *Proceedings of the 7th IEEE conference on industrial electronics and applications* (pp. 1999–2003).

Hai, L. Q., Somjit, A., & Ngamnij, A. (2012). Association rule hiding based on distance and intersection lattice. In *Proceedings of the 4th international conference on computer technology and development* (pp. 227–231).

Hai, L. Q., Somjit, A., & Ngamnij, A. (2013a). Association rule hiding based on intersection lattice. *Mathematical Problems in Engineering*.

Hai, L. Q., Somjit, A., Huy, X. N., & Ngamnij, A. (2013b). Association rule hiding in risk management for retail supply chain collaboration. *Computers in Industry, 64*, 776–784.

Hajian, S., Domingo-Ferrer, J., & Farràs, O. (2014). Generalization-based privacy preservation and discrimination prevention in data publishing and mining. *Data Mining and Knowledge Discovery, 28*, 1158–1188.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. In *Proceedings of the international conference on management of data* (pp. 1–12).

Han, J., Pei, J., Yin, Y., & Mao, R. (2004). Mining frequent pattern without candidate generation: A frequent pattern tree approach. *Data Mining and Knowledge Discovery, 8*(1), 53–87.

Harik, G. R., Lobo, F. G., & Goldberg, D. E. (1999). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation, 3*(4), 287–297.

Holland, J. H. (1992). *Adaptation in natural and artificial systems*. MIT Press.

Hong, T. P., Lin, C. W., Yang, K. T., & Wang, S. L. (2013). Using TF-IDF to hide sensitive itemsets. *Applied Intelligence, 38*(4), 502–510.

Hong, T. P., Wang, C. Y., & Tao, Y. H. (2001). A new incremental data mining algorithm using pre-large itemsets. *Intelligent Data Analysis, 5*, 111–129.

Huang, H., Wu, X., & Relue, R. (2002). Association analysis with one scan of databases. In *Proceedings of the IEEE international conference on data Mining* (pp. 629–632).

Kennedy, J., & Eberhart, R. (1995). Particles warm optimization. In *Proceedings of the IEEE international conference on neural networks* (pp. 1942–1948).

Kowalski, G. J., & Maybury, M. T. (2006). *Information storage and retrieval systems: Theory and implementation*. Springer.

Kumar, V. (1992). Algorithms for constraint-satisfaction problems: A survey. *AI Magazine, 13*(1).

Kuo, R. J., Chao, C. M., & Chiu, Y. T. (2011). Application of particle swarm optimization to association rule mining. *Applied Soft Computing, 11*(1), 326–336.

Li, J., Shen, H., & Topor, R. (2001). Mining the smallest association rule set for predictions. In *Proceedings of the IEEE international conference on data mining* (pp. 361–368).

Li, Y. C., & Chang, C. C. (2007). MICF: An effective sanitization algorithm for hiding sensitive patterns on data mining. *Advanced Engineering Informatics, 21*, 269–280.

Lin, C. W., Hong, T. P., Wong, J. W., Lan, G. C., & Lin, W. Y. (2014b). A GA-based approach to hide sensitive high utility itemsets. *Scientific World Journal, 2014*. doi:10.1155/2014/804629.

Lin, C. W., Hong, T. P., Yang, K. T., & Wang, S. L. (2015). The GA-based algorithms for optimizing hiding sensitive itemsets through transaction deletion. *Applied Intelligence, 42*(2), 210–230.

Lin, C. W., Liu, Q., Fournier-Viger, P., Hong, T. P., Voznak, M., & Zhan, J. A. (2016). A sanitization approach for hiding sensitive itemsets based on particle swarm optimization. *Engineering Applications of Artificial Intelligenc, 53*, 1–18.

Lin, C. W., Zhang, B., Yang, K. T., & Hong, T. P. (2014a). Efficiently hiding sensitive itemsets with transaction deletion based on genetic algorithms. *Scientific World Journal, 2014*. doi:10.1155/2014/398269.

Lin, J. L., & Cheng, Y. W. (2009). Privacy preserving itemset mining through noisy items. *Expert Systems with Applications, 36*, 5711–5717.

Lin, J. L., & Liu, J. Y. C. (2007). Privacy preserving itemset mining through fake transactions. In *Proceedings of the 22nd annual ACM symposium on applied computing*.

Lin, W., Alvarez, S., & Ruiz, C. (2002). Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery, 6*, 83–105.

Lindell, Y., & Pinkas, B. (2009). Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality, 1*(1), 59–98.

Liu, Y., Liao, W. K., & Choudhary, A. (2005). A two-phase algorithm for fast discovery of high utility itemsets. In *Proceedings of the pacific-asia conference on knowledge discovery and data mining* (pp. 689–695).

Luenberger, D. (1973). *Introduction to linear and Non-linear programming*. Addison-Wesley Publishing Company.

Maimon, O., & Rokach, L. (2010). *Data mining and knowledge discovery handbook*. Springer.

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

Menon, S., & Sarkar, S. (2008). Minimizing information loss and preserving privacy. *Manage Science, 53*, 101–116.

Menon, S., Sarkar, S., & Mukherjee, S. (2005). Maximizing accuracy of shared databases when concealing sensitive patterns. *Information Systems Research, 16*(3), 256–270.

Moskowitz, I., & Chang, L. (2000). A decision theoretic system for information downgrading. In *Proceedings of the joint conference on information sciences*.

Moustakides, G. V., & Verykios, V. S. (2006). A max–min approach for hiding frequent itemsets. In *Proceedings of the 6th IEEE international conference on data mining* (pp. 502–506).

Moustakides, G. V., & Verykios, V. S. (2008). A MaxMin approach for hiding frequent itemsets. *Data & Knowledge Engineering, 65*, 75–89.

O'Leary, D. (1991). Knowledge Discovery as a Threat to Database Security. In G. Piatetsky-Shapiro, & W. J. Frawley (Eds.), *Knowledge discovery in databases* (pp. 507–516). Menlo Park: AAAI/MIT Press.

O'Leary, D. E. (1995). Some privacy issues in knowledge discovery: The OECD personal privacy guidelines. *IEEE Expert, 10*(2), 48–52.

O'Mahony, M., Hurley, N., Kushmerick, N., & Silvestre, G. (2004). Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology, 4*(4), 344–377.

Oliveira, S. R. M., & Zaiane, O. (2004). Data perturbation by rotation for privacy-preserving clustering. *Technical Report TR04-17*.

Oliveira, S. R. M., & Zaiane, O. R. (2002). Privacy preserving frequent itemset mining. In *Proceedings of the IEEE international conference on privacy, security and data mining* (pp. 43–54).

Oliveira, S. R. M., & Zaiane, O. R. (2003a). Protecting sensitive knowledge by data sanitization. In *Proceedings of the 3rd IEEE international conference on data mining* (pp. 211–218).

Oliveira, S. R. M., & Zaiane, O. R. (2003b). Algorithms for balancing privacy and knowledge discovery in association rule mining. In *Proceedings of the international database engineering and applications symposium* (pp. 54–63).

Oliveira, S. R. M., & Zaiane, O. R. (2006). A unified framework for protecting sensitive association rules in business collaboration. *International Journal of Business Intelligence and Data Mining, 1*(3), 247–287.

Oliveira, S. R., & Zaiane, O. R. (2010). Privacy preserving clustering by data transformation. *Journal of Information and Data Management, 1*(1), 37.

Pontikakis, E. D., Tsitsonis, A. A., & Verykios, V. S. (2004a). An experimental study of distortion-based techniques for association rule hiding. In *Proceedings of the 18th conference on database security* (pp. 325–339).

Pontikakis, E. D., Theodoridis, Y., Tsitsonis, A. A., Chang, L., & Verykios, V. S. (2004b). A quantitative and qualitative analysis of blocking in association rule hiding. In *Proceedings of the ACM workshop on privacy in the electronic society* (pp. 29–30).

Prakash, M., & Singaravel, G. (2015). An approach for prevention of privacy breach and information leakage in sensitive data mining. *Computers and Electrical Engineering, 45*, 134–140.

Reddy, M. R., & Wang, R. Y. (1995). Estimating data accuracy in a federated database environment. In *Proceedings of the 6th international conference of information systems and data management* (pp. 115–134).

Rizvi, S. J., & Haritsa, J. R. (2002). Maintaining data privacy in association rule mining. In *Proceedings of the 28th conference on very large databases* (pp. 682–693).

Russell, S., & Norvig, P. (2003). *Artificial Intelligence: a modern approach*: 27. Prentice-Hall.

Salton, G., Fox, E. A., & Wu, H. (1983). Extended Boolean information retrieval. *Communications of the ACM, 26*(11), 1022–1036.
Samarati, P. (2001). Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering, 13*(6), 1010–1027.
Saygin, Y., Verykios, V. S., & Clifton, C. (2001). Using unknowns to prevent discovery of association rules. *ACM SIGMOD Record, 30*(4), 45–54.
Saygin, Y., Verykios, V. S., & Elmagarmid, A. K. (2002). Privacy preserving association rule mining. In *Proceedings of the international workshop on research issues in data engineering: Engineering E-commerce/E-business systems* (pp. 151–158).
Sohrabi, M. K., & Roshani, R. (2017). Frequent itemset mining using cellular learning automata. *Computers in Human Behavior, 68*, 244–253.
Sun, X., & Yu, P. S. (2005). A border–based approach for hiding sensitive frequent itemsets. In *Proceedings of the 5th IEEE international conference on data mining* (pp. 426–433).
Sun, X., & Yu, P. S. (2007). Hiding sensitive frequent itemsets by a border-based approach. *Computing Science and Engineering, 1*(1), 74–94.
Sweeney, L. (2002). K-Anonymity: A model for protecting privacy. *International Journal Uncertain Fuzziness Knowledge Based Systems, 10*(5), 557–570.
Telikani, A., & Shahbahrami, A. (2017). Optimizing association rule hiding using combination of border and heuristic approaches. *Applied Intelligence, 47*, 544–557.
Verykios, V. S. (2013). Association rule hiding methods. *WIREs Data Mining and Knowledge Discovery, 3*, 28–36.
Verykios, V. S., & Divanis, A. G. (2008). Survey of association rule hiding methods for privacy. In C. C. Aggarwal, & P. S. Yu (Eds.), *Privacy-Preserving data Mining: models and algorithms* (pp. 267–289). New York: Springer.
Verykios, V. S., Bertino, E., Fovino, I. N., Parasiliti, L., Saygin, Y., & Theodoridis, Y. (2004a). State-of-the-art in privacy preserving data mining. *ACM SIGMOD Record, 33*(1), 50–57.
Verykios, V. S., Elmagarmid, A. K., Bertino, E., Saygin, Y., & Dasseni, E. (2004b). Association rule hiding. *IEEE Transactions on Knowledge and Data Engineering, 16*(4), 434–447.
Verykios, V. S., Pontikakis, E. D., Theodoridis, Y., & Chang, L. (2007). Efficient algorithms for distortion and blocking techniques in association rule hiding. *Distributed and Parallel Databases, 22*(1), 85–104.
Wang, S. L. (2009). Maintenance of sanitizing informative association rules. *Expert Systems with Applications, 36*, 4006–4012.
Wang, S. L., & Jafari, A. (2005). Using unknowns for hiding sensitive predictive association rules. In *Proceedings of the IEEE international conference on information reuse and integration* (pp. 223–228).
Wang, S. L., Maskey, R., Jafari, A., & Hong, T. P. (2008). Efficient sanitization of informative association rules. *Expert Systems with Applications, 35*(1-2), 442–450.
Wang, S. L., Parikh, B., & Jafari, A. (2007a). Hiding informative association rule sets. *Expert Systems with Applications, 33*(2), 316–323.
Wang, S. L., Patel, D., Jafari, A., & Hong, T. P. (2007b). Hiding collaborative recommendation association rules. *Applied Intelligence, 26*(1), 66–77.
Wu, Y. H., Chiang, C. M., & Chen, A. L. (2007). Hiding sensitive association rules with limited side effects. *IEEE Transactions on Knowledge and Data engineering, 19*(1).
Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. In *Proceedings of the IEEE nature & biologically inspired computing* (pp. 210–214).
Yao, A. C. (1982). Protocol for secure computations. In *Proceedings of the 23rd annual IEEE symposium on foundation of computer science* (pp. 160–164).
Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering, 12*(3), 372–390.
Zeng, Y., Yin, S., Liu, J., & Zhang, M. (2015). Research of improved FP-growth algorithm in association rules mining. *Scientific Programming*.
Zheng, Z., Kohavi, R., & Mason, L. (1999). Real world performance of association rule algorithms. In *Proceedings of the 7th ACM SIGKDD international conference of knowledge discovery data mining* (pp. 401–406).