Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2014) 22: 434 – 450

doi:10.3906/elk-1206-66

Privacy preserving in association rules using a genetic algorithm

Rahat Ali SHAH[1,*], **Sohail ASGHAR**[2]
[1]International Islamic University, Islamabad, Pakistan
[2]UIIT, PMAS- Arid Agriculture University, Rawalpindi, Pakistan

Received: 15.06.2012 • **Accepted:** 03.12.2012 • **Published Online:** 17.01.2014 • **Printed:** 14.02.2014

Abstract: Association rule mining is one of the data mining techniques used to extract hidden knowledge from large datasets. This hidden knowledge contains useful and confidential information that users want to keep private from the public. Similarly, privacy preserving data mining techniques are used to preserve such confidential information or restrictive patterns from unauthorized access. The pattern can be represented in the form of a frequent itemset or association rule. Furthermore, a rule or pattern is marked as sensitive if its disclosure risk is above a given threshold. Numerous techniques have been used to hide sensitive association rules by performing some modifications in the original dataset. Due to these modifications, some nonrestrictive patterns may be lost, called lost rules, and new patterns are also generated, known as ghost rules. In the current research work, a genetic algorithm is used to counter the side effects of lost rules and ghost rules. Moreover, the technique can be applied for small as well as for large datasets in the domain of medical, military, and business datasets.

Key words: Privacy preserving data mining, association rules, sensitive association rules hiding, genetic algorithm

1. Introduction

Organizations such as customer relationship management, the telecommunication industry, financial sector investment trends, web technologies, demand and supply analysis, direct marketing, health industry, e-commerce, stocks and real estate, understanding consumer research marketing, e-commerce, and product analysis generate huge amounts of data that often contain useful information (i.e. name, address, age, salary, social security number, type of disease, and the like). Through data mining, we are able to extract useful and previously unknown information that organizations or individuals do not want to disclose to the public. Therefore, privacy preserving data mining (PPDM) techniques are applied to preserve such confidential information from any type of mining algorithm [1–6]. Hence, the basic objective of PPDM is to protect data against serious adverse effect. In addition, the privacy regarding data mining is divided into 2 types. The first type of privacy, termed as output privacy, is where the data are altered so that the mining result will conserve certain privacy. Many modification techniques such as perturbation, blocking, aggregation, swapping, and sampling are used for this type of privacy [7–15]. The second type of privacy, labeled as input privacy, is where the data are manipulated so that the mining result is not affected or is less affected. Cryptography- and reconstruction-based techniques are used for this type of privacy [16–20].

The mining association rule (AR) is a 2-step process. In step 1, the Apriori algorithm is used to mine frequent k-itemsets [21] from huge amounts of data. In step 2, ARs are generated from frequent k-itemsets.

*Correspondence: rahatalishah_b@yahoo.com

Furthermore, a rule is called sensitive if its discloser risk is above a user-specified threshold. In addition, sensitive rules contain confidential data that we do not want to disclose to the public. Example: consider 2 retailers, Bob and Ali, in a supermarket. Bob is the experienced one and Ali has only recently joined the market. Now, Ali wants to place those items or products that the customers purchase more or whose purchase ratio is high. For this purpose, he wants to see Bob's ARs, such as any customer of Bob's who buys milk as well as tea. We call this rule sensitive for Bob. Similarly, if Ali knows of any customer of Bob's who buys milk as well as tea, he runs a coupon scheme that offers some discounts on milk with the purchase of tea. Gradually, Bob's sale of milk with tea decreases and Ali's sales increase. Consequently, Ali monopolizes the market, as shown in Figure 1.

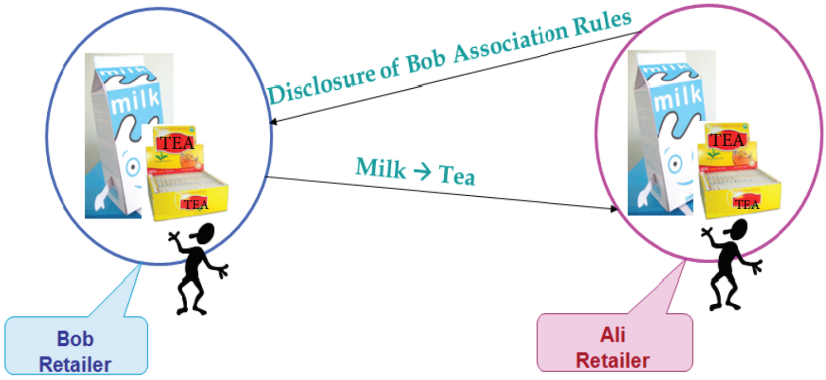

Figure 1. PPDM supermarket example.

ARs can be divided into 2 subcategories: weak ARs (WARs) and strong ARs (StARs). A rule is called a WAR if its confidence is lower than the user-specified threshold. Similarly, a rule is marked as a StAR if its confidence is greater than or equal to the user-specified threshold. Moreover, StARs are further divided into sensitive ARs (SARs) and nonSARs. Furthermore, 2 strategies are used to hide SARs [22].

- Increase support of the antecedent.
- Decrease support of the consequent.

This work is based on support and confidence framework. The support is the measure of the occurrences of a rule in a transactional database, while the confidence is a measure of the strength of the relation between sets of items. An AR is an implication of the form $X \Rightarrow Y$ where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \emptyset$, where $I = \{i_1, i_2, i_3 \ \ldots I_m\}$ is a set of literals, called items. X is called the body or antecedent (tail) of the rule and Y is called the head or consequent of the rule. An example of such a rule is when 60% of the customers who buy bread also buy butter. The confidence of the rule will be 100%, which means that 60% of the records that contain bread also contain butter. The confidence of the rule is the number of records that contain both the left-hand side (X) and right-hand side (Y), divided by the number of records that contain the left-hand side (X), which is calculated with the following formula:

$$\text{Confidence } (X \Rightarrow Y) = \frac{|X \cup Y|}{|X|}. \tag{1}$$

The support of the rule is the percentage of transaction that contains both the left-hand side (X) and

right-hand side (Y), which is calculated with the following formula:

$$\text{Support } (X \Rightarrow Y) = \frac{|X \cup Y|}{|Y|}, \text{ where } N \text{ is the number of transactions in } D. \quad (2)$$

A lot of research has been done in the domain of PPDM. Most of the existing techniques are based on the support and confidence framework. In addition, we identify that these techniques are suffering from the side effects of lost rules, some lost nonrestrictive patterns and ghost rules, and some new patterns are falsely generated, which may not be supported by the original database. These side effects play an important role in the motivation of the proposed architecture. In current research work, a genetic algorithm (GA) is used to improve the existing PPDM in the domain of the lost rule and ghost rule side effects. Moreover, the GA is an evolutionary and metaheuristic technique used to solve complex problems. Hence, preserving the privacy of ARs is a complex problem and needs optimal sanitization. Therefore, the GA is used to provide the optimal solution to such a hard problem. In the same direction, a new modification technique is introduced, called the privacy preserving GA (PPGA). This technique modifies the database recursively until the support or confidence of the restrictive patterns drop below the user-specified threshold. Furthermore, the technique is only applicable on binary datasets. In addition, the technique introduced in this paper only modifies those transactions that contain the maximum number of sensitive items and minimum number of availability of nonsensitive items. The technique can be applicable for small datasets as well as for large datasets. In order to test and validate the performance of the PPGA-based framework, experiments are conducted on the Zoo, Synthetic, and Extended Bakery datasets. On the basis of the experimental results, the claim is validated that the proposed framework gives better results than the existing state of the art techniques based on the rule hiding distance, not on lost and ghost rules.

The work done in this paper can be divided into 3 phases. In phase 1, k-frequent itemsets are generated and then ARs are generated from these itemsets. PPGA is applied to release a sanitized database in order to hide SARs in the second phase. In phase 3, the original database is compared to the sanitized database, to find the number of lost rules and ghost rules.

The format of this paper is as follows: Section 2 is a literature review, which describes the comprehensive study on SARs hiding or PPDM in ARs. Section 3 presents the proposed model for PPDM, components of PPGA, and its implementation. In Section 4 the experimental results of the proposed technique are compared with other techniques existing in the literature. Section 5 describes the conclusions and future work.

2. Literature review

Sharing data is often beneficial but sometimes discloses confidential information. PPDM techniques are used to preserve confidential information from unauthorized access. In this paper, we focus on issues regarding privacy preserving in ARs (PPARs). In this context, we review the literature in order to analyze and find limitations in the existing literature. Aggarwal and Yu [23] proposed that preserving the privacy of restrictive patterns refers to the process of modifying the original database in such a way that some restrictive patterns hide without seriously affecting the data and the nonrestrictive patterns. Generally, the process of modification can be divided into data blocking and data distortion techniques. The major concept of data distortion techniques is the replacement of selected values with 'false' values (i.e. replacing 1s with 0s and vice versa). Moreover, this technique is applicable to reduce the support or confidence of the SARs from the user-specified threshold. Analyses concerning the use of this technique can be found in Verykios et al. [24], Duraiswamy et al. [25], and Dehkordi et al. [26]. All of these approaches add false values to the real transaction, which causes the

problems of lost rule and ghost rule side effects. Similarly, the major concept of a blocking technique is the replacement of an existing attribute value with 'unknown' or '?'. In a blocking technique, the algorithms do not add a false value to the database. In addition, to restore a value by an unknown value instead of placing a false value is slightly more advantageous for a specific application such as a medical application. Examinations regarding the use of this technique can be found in Dasseni et al. [11], Weng et al. [27], and Saygin et al. [15,28]. The solution presented in these approaches uses a blocking method to transform the original database D into the release database $D`$, by increasing the support of the rule antecedent by changing 0s to ? or by decreasing the support of the rule consequent by changing 1s to ?. Hence, compared to other techniques in the literature, blocking-based techniques do not distort the database, they only change some of the known values to unknown. The main limitation of this technique is the privacy violation of the modified database. For example, the opponent can easily leak the information by replacing the question mark by 1s or 0s.

In the same direction, Clifton et al. [29] discussed the security issues and implication of data mining. He did not propose any specific algorithm. Moreover, he investigated the idea of limiting access to the database; supplementing data, remove needless combinations, audit and fuzzy data. Later on, Atallah et al. [30] proved that optimal sanitization is a nondeterministic polynomial time (NP)-hard problem and needs standardization. In their research, they proposed a heuristic based on support reduction, to exclude sensitive frequent itemsets. In a similar direction, Verykios et al. [24] introduced 5 algorithms. Generally, these algorithms run on the strategy that is based on reducing the support and confidence of rules. Moreover, here, distortion is used as a modification technique. More precisely, none of these techniques are the best to overcome all of the side effects caused by preserving the privacy of ARs. Similarly, the time taken by each algorithm to hide a set of rules is also high. Chih-Chia et al. [31] proposed a novel algorithm, the fast hiding SARs (FHSARs). Typically, the technique hides SARs successfully by establishing a correlation between the transaction and SARs. Generally, the proposed technique assigns a weight W to each transaction. The weight shows the dependency of the transaction on restrictive patterns. Moreover, it generates fewer new rules than the previous work done. The bottleneck of FHSARs is the number of lost rules and performance in terms of W, which is computed again after each item is modified. In the same direction, Dehkordi et al. [26] used a GA in the domain of privacy preserving in ARs. In addition, the technique divided the database into safe and critical transactions. Safe transactions are those that do not contain sensitive items and there is no need to sanitize. While critical transactions are those that contain sensitive items and need to be sanitized. The solution presented in this approach uses the distortion (replacing 1s by 0s and vice versa) method to transform the original database D into the release database $D`$. Dehkordi et al. did not define a dataset for experimentation. Moreover, the side effect in terms of the lost and ghost rules were not clearly defined. Furthermore, the technique was not compared to the previous techniques existing in the literature. Moreover, in the support and confidence of ARs more recently, Naeem et al. [32] proposed a novel architecture in the domain of PPDM. In this approach, the authors used 5 measures, namely confidence, all-confidence, conviction, leverage, and lift, in order to mine ARs from large databases. In addition, the author used a weighting mechanism, which was used to assign a weight to each transaction. The weighting mechanisms used in this approach are, sum, mean, median, and mode. Consequently, the technique is only applicable on datasets whose attributes are not more than 26. The technique generates 0 ghost rule side effects. Furthermore, the technique generates high lost rule side effects.

3. Proposed model for the PPDM

It is clear from the literature that most of the techniques suffer from the side effects of lost rules and ghost rules. These limitations play an important role in the motivation of the proposed architecture. In the current

research work, the GA is used to counter the side effects of lost rules and ghost rules. This work has a partial resemblance to the work done by Dehkordi et al. [26]. However, the difference is that we define our own fitness strategy. The question arises, "Why are we using a GA in PPDM?" The answer to such a question is that PPDM is an extremely complex domain and needs to be standardized [29]. Such standardization in PPDM refers to a NP-hard problem [30]. Therefore, a GA is used to provide the optimal solution to the hard problem. Such optimality of the solution depends on the complexity of the fitness function. The possible strength of the fitness function ensures a desirable level of the optimal solution. A GA was developed by Holland in [33]. Holland's GA is a method for moving from one population of 'chromosomes' (e.g., strings of 'bits' representing candidate solutions to a problem) to a new population. In terms of the GA, the dataset is called a population and the transaction is called a chromosome. Moreover, a GA is an evolutionary and metaheuristic technique used to solve complex problems. Therefore, a GA is used to hide restrictive patterns, $X \rightarrow Y$, by decreasing the support of Y or by increasing the support of X. Furthermore, it often requires a 'fitness function'. Hence, the fitness function assigns a value to each transaction (chromosome) in the database (population). Additionally, the fitness of the transaction depends on how well that transaction solves the problem at hand. The fitness is calculated with the help of Eqs. (5) and (8).

Let D be a set of transactions in a dataset, denoted as $D = \{T_1, T_2, \ldots\ldots\ldots, T_n\}$ and let R be a set of identifiers, defined as $R = \{1, 2, \ldots\ldots\ldots, n\}$. Each record T_r is defined as a set of data items, $Tr = \{d_1, d_2, \ldots\ldots\ldots\ldots d_k\}$, where I represents a set of identifiers, $I = \{1, 2, \ldots\ldots\ldots\ldots\ldots, k\}$.

Let S be a set of sensitive items or a sensitive pattern, denoted as $S = \{s1, s2, \ldots\ldots\ldots\ldots, sm\}$ and let P be a set of identifiers for the elements of S, defined as $P = \{1, 2, \ldots\ldots\ldots\ldots, m\}$.

$$\forall S_p \in T_r \bigvee S_p \notin T_r, 1/\sum_{i=1}^{k} Count\,(S_p)\ in\, T_r : S_p \geq 1 \tag{3}$$

e.g.: D = {T_1, T_2, T_3}
T1 = {Bread, Butter}
T_2 = {Bread, Egg}
T3 = {Bread, Butter}
S_p = {Bread, Butter}
Count (S_p) in T_2 = (Bread)
$S_p \in T_2, \sum_{i=1}^{k} Count\,(S_p)\ in\, T_2 = 1$
$1/\ \sum_{i=1}^{k} Count\,(S_p)\ in\, T_2 = (\frac{1}{1})$

$$\forall d_i \in T_r, \sum_{i=1}^{k} d_i : d_i \rightarrow [1] \tag{4}$$

e.g.: $d_i \in T_2$, $\sum_{i=1}^{k} d_i = (1 + 1) = 2$

Let F be a set of fitness values, defined as $F = \{f_1, f_2, \ldots\ldots\ldots f_h\}$, and let V be a set of identifiers for the elements of F, denoted as $V = \{1, 2, \ldots\ldots\ldots h\}$.

$$\forall f_v \in T_r, \text{ where } f_v = 1/\sum_{i=1}^{k} Count\,(S_p)\ in\, T_r + \sum_{i=1}^{n} d_i \tag{5}$$

e.g.: $\sum_{i=1}^{k} d_i = 2$

$\sum_{i=1}^{k} Count\,(S_p)\ in\, T_2 = 1$

$\mathrm{f}_v\ = (1/1) + 2 = 3$

Eq. (5) shows that the fitness value depends on the number of sensitive items in a transaction. This means that the fitness function is rule-oriented. Moreover, transactions are sorted in descending order on the basis of the fitness value. Furthermore, the transaction having a lower fitness value will be selected for modification. Hence, the fitness function goes toward maximization.

Let C be a set of chromosomes, denoted as $C = \{O_1, O_2, \ldots\ldots\ldots, O_n\}$, and let S be a set of identifiers, defined as $S = \{1, 2, \ldots\ldots\ldots, n\}$. Each record O_s is defined as a set of items, $O_s = \{o_1, o_2, \ldots\ldots\ldots\ldots o_k\}$, where I represents a set of identifiers, $I = \{1, 2, \ldots\ldots\ldots\ldots\ldots, k\}$.

$|D| = |C|\ |\forall |O_s| = |T_r|$

$\mathrm{D}: = \mathrm{C}\ \wedge\ \mathrm{T}_r : = \mathrm{O}_s$

$$S_p \in O_s \forall S_p \notin O_s\,, 1/\ \sum_{i=1}^{k} Count\,(S_p)\ in\, O_s : S_p \geq 1 \tag{6}$$

$$o_i \in O_s, \sum_{i=1}^{k} o_i : o_i \rightarrow [0] \tag{7}$$

The fitness for each offspring can be calculated by Eq. (8).

$$f_v \in O_s,\ \text{where}\ f_v = 1/\sum_{i=1}^{k} Count\,(S_p)\ in\, O_s + \sum_{i=1}^{n} o_i \tag{8}$$

The question arises, "How do we justify the fitness function?" The answer to such a question is that the fitness function is divided into 2 parts. The first part is called transaction sensitivity; it increases the priority by decreasing the value of those transactions that contain sensitive items, as shown in Eqs. (3) and (6). On the basis of these equation transactions, those having the maximum number of sensitive items will be selected for modification. The second part is called transaction priority; it increases the priority of selected transactions containing the same number of sensitive items, as shown in Eqs. (4) and (7). In other words, we can say that on the basis of Eq. (5), those transactions will be selected for modifications (to replace with offspring) that contain the maximum number of sensitive items (availability of data items) and minimum number of data items. In doing so, the lost rule side effect is minimized. Similarly, on the basis of Eq. (8), those offspring will be selected to be replaced with transactions in the original database that contain the minimum number of sensitive items and maximum number of nonavailability of the data items. By doing this, the ghost rule side effect is minimized.

Definition 1 *PPGA hides SARs successfully.*

$$\exists S_p \in T \Delta T_r | \exists ! S_p \in T_r \tag{9}$$

Eq. (9) shows that the proposed technique hides SARs successfully, because the technique only modifies those transactions in which sensitive items are present.

Definition 2 *PPGA minimizes lost rule side effects.*

$$S_p \in T_r \wedge f_v \in T_r | T_r \Delta O_s f_v \in T_r : f_{v \leq} f_{v+1} \tag{10}$$

Eq. (10) shows that the lost rule is minimized because the technique selects those transactions to modify in which fewer data items are available.

Definition 3 *PPGA minimizes ghost rule side effects.*

$$S_p \in O_s \wedge f_v \in O_s | T_r = O_s f_v \in O_s : f_v \geq f_{v+1} \tag{11}$$

Eq. (11) illustrates that the ghost rule is minimized because the selected transactions are replaced by those offspring in which the maximum number of data items are unavailable.

The notation used in the proposed architecture is shown in Table 1. Figure 2 shows the element organization of the PPGA for hiding restrictive patterns.

Table 1. Notation and definition.

Notation	Details
D	Original dataset
$D`$	Sanitized dataset
T_r	Transaction ID
ARs	Association rules
SARs	Sensitive ARs
MCT	Minimum confidence threshold
MST	Minimum support threshold
f_v	Fitness of each transaction (chromosome)
TMG	Transaction modified in each generation
O_s	Offspring ID
LRs	Lost rules
GRs	Ghost rules

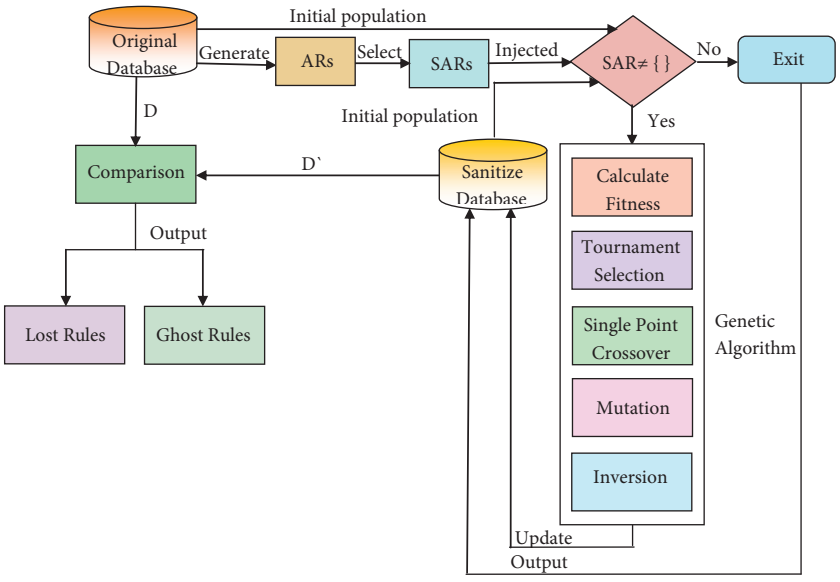

Figure 2. Framework of the PPGA for hiding SARs.

3.1. Components of the PPGA

The components of the proposed model are divided into 3 phases, as shown in Figure 3.

1. Input: Original Database D, SARs, MCT, MST, N, Replace
2. Output: Transform D into D'

Phase -1

3. FS → Frequent Itemset (D)
4. AR → Generate Association Rules (FS)
5. SAR → Select Sensitive Association Rules (AR)

Phase -2

6. WHILE SAR{} != Ø OR generation != N
7. Fitness: $f_v=1/\sum_{i=1}^{k}$ Count(Sp) in $T_r+\sum_{i=1}^{k} d_i$: $d_i \rightarrow [1]$
8. Selection: Base on f_v
9. Crossover: $T_r * T_{r+1}$
10. Mutation: Select T_r, Change 1 to 0 or 0 to 1 randomly
11. Fitness: $f_v= 1/\sum_{i=1}^{k}$ Count(Sp) in $O_s+\sum_{i=1}^{k} o_i$: $o_i \rightarrow [0]$
12. Replace: $T_r \Delta O_s$
13. Wend

Phase -3

14. D $\oplus$ D`

Figure 3. PPGA.

In phase 1, the data are first converted into Boolean format, such as 1 and 0, where 1 represents the availability of data items and 0 shows the nonavailability of data items. Next, the Apriori algorithm is applied with some minimum support thresholds (MSTs) to mine k-frequent itemsets, and then ARs are generated from these frequent itemsets. In addition, we select some of these ARs as SARs, whose confidences are greater than or equal to the minimum confidence threshold (MCT).

In phase 2, the PPGA transforms the original database D into the sanitized database D`, such that none of the SARs are derived. At first, the confidence of the SAR is obtained from the original dataset (initial population) and compared with the MCT. If the confidence of the SAR is greater than or equal to the MCT, then the fitness of each transaction is calculated by Eq. (3). Next, different operators of the PPGA are applied. This phase of the PPGA runs repeatedly until SAR{} ! = Ø.

In phase 3, the original database D is compared to the modified database D`, to find the number of ghost rule and lost rule side effects.

Tournament selection: In the tournament selection, 2 chromosomes are selected randomly from the population and the most fit of these 2 is selected for the mating pool, as shown in Table 2.

Table 2. T-selection.

Population	Fitness	T-selection	
11100	3.5	10010 10100	10010
10100	2.5		
10010	3.0		
01011	3.0		

Single point crossover: In the single point crossover, the parent's chromosome is split into 2 portions, such as the head and tail. Similarly, the head of one chromosome combines with the tail of another chromosome in the mating pool, as illustrated in Figure 4.

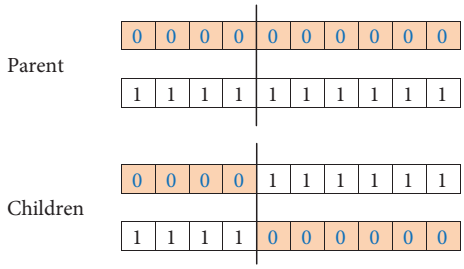

Figure 4. One-point crossover.

Mutation operator: The mutation operator randomly changes the values (1 to 0 or 0 to 1) of some locations in the chromosome, as depicted in Figure 5.

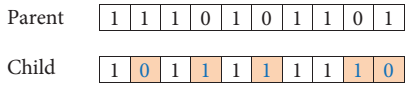

Figure 5. Mutation.

Replacement or inversion operator: In this operator, some of the chromosomes of the initial population will be replaced with some of the chromosomes of the offspring, as depicted in Table 3. The fitness values are calculated by Eq. (8).

Table 3. Inversion operation.

Population	Offspring	Mutation	Fitness	Inversion
11100	10100	00100	5.0	11100
10100	11010	11110	1.5	11000
10010	11100	10100	3.5	10010
01011	11100	11000	4.0	01011

Example: Table 4 shows the example dataset in comma-separated values (CSV) file format. The example dataset contains 4 transactions and 5 items in each transaction.

Table 4. Example dataset.

Transaction ID	Items bought
1	Bread, Butter, Egg
2	Bread, Egg
3	Bread, Tea
4	Butter, Tea, Cake

If the MST is 50%, then $\{Bread,\ Egg\}$ is the only 2-itemset that satisfies the minimum support, as shown in Table 5. Thus, if the MST is increased from 50%, then the information of $\{Butter\}$ and $\{Egg\}$ is lost, which affects the association of $\{Bread,\ Egg\}$. Hence, if the MST is decreased, then the ratio of unuseful information is increased.

If the MCT is 50%, then only 2 rules are generated from this 2-itemset, which have confidences of greater than 50%, as shown in Table 6. As we know, ARs are generated from the frequent itemset; therefore, the MCT must be greater than or equal to the MST. In the current example, if the MCT is 66%, then the same result will be shown. Similarly, if the MCT is 67%, then rule $Bread \rightarrow Egg$ is lost, which we want to hide.

Table 5. Frequent itemset.

Frequent itemset	Support
{Bread}	75%
{Butter}	50%
{Egg}	50%
{Tea}	50%
{Bread, Egg}	50%

Table 6. ARs from the frequent itemset.

Antecedent	→	Consequent	Support	Confidence
Bread	→	Egg	50%	66%
Egg	→	Bread	50%	100%

Assume that rule *Bread* → *Egg* is sensitive or leaks confidential information and needs to be hidden. At first, some input parameters, such as the initial population, MST, MCT, and SAR, are passed to the PPGA. Later on, the PPGA calculates the fitness for each transaction. On the basis of fitness, transactions are selected for the new generation. Hence, the fitness generated by one rule is deferred from another rule. However, the fitness depends on the rule or is rule oriented. Table 7 describes the fitness of rule *Bread* → *Egg* generated from example dataset.

Table 7. Fitness of the example dataset.

Population	Fitness
11100	3.5
10100	2.5
10010	3.0
01011	3.0

Table 8 demonstrates the first iteration or generation of the PPGA. In the next generation, the sensitivity of the rule is checked by comparing the confidence and support of the rule to the user-specified threshold. If the sensitivity of the rule is below the specified threshold, it means that the rule is hidden. Subsequently, the modified dataset is compared to the original dataset to achieve the lost rule and ghost rule side effects.

Table 8. First iteration of the PPGA.

T-selection		X-over	Offspring	Mutation	Fitness	Inversion
10100 10010	10010	10|010	10100	00100	5.0	11100
11100 10010	10010	11|100	11010	11110	1.5	00100
11100 10100	11100	11|100	11100	10100	3.5	10010
11100 10100	11100	11|100	11100	11000	4.0	01011

Table 9 illustrates that after a single iteration of the PPGA, the support and confidence of rule *Bread* → *Egg* drops below the user-specified threshold. Thus, the proposed technique hides SARs successfully.

Due to this modification, rule *Egg*→*Bread* is falsely hidden, which is called a lost rule, and no new rule is generated.

Table 9. Performance measure of the PPGA.

Performance measure	AR	Support	Confidence
SAR	Bread→Egg	25%	50%
Lost rule	Egg→Bread	25%	50%

3.2. Implementation

In this section, the screen shot of the PPGA is given. The development of the PPGA is coded in Java, using NetBeans IDE 6.9.1 as a development tool. Java is selected as a programming language because of its prominent features. Moreover, it provides a high performance graphical user interface. In addition, we perform a series of experiments on a PC with a Core i3 ∼2.1-GHz central processing unit (CPU) and 4-GB memory, under the Windows 7 Professional 64-bit operating system. At first, the data that are in the CSV file format are imported

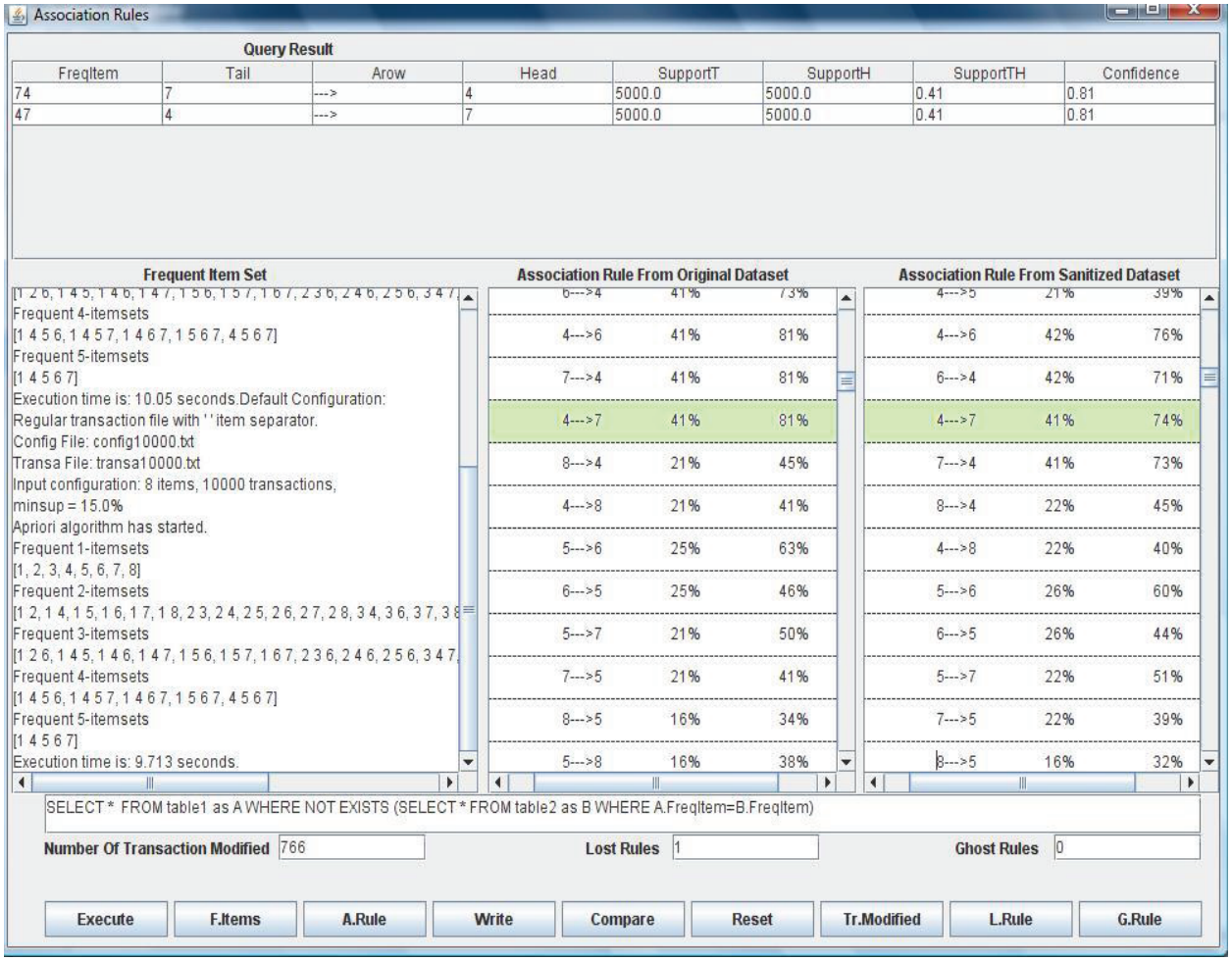

Figure 6. Mining frequent itemsets and ARs from the synthetic dataset.

to the MySQL database. After this, the Apriori algorithm is used to mine the frequent k-itemset. Figure 6 describes the ARs generated from frequent k-itemsets before and after sanitization from the synthetic dataset, which will be discussed later on. It indicates the SAR 4→7, with a support of 41% and confidence of 81%. The rule is hidden by decreasing the confidence to 74%, as shown in the column of the ARs from the sanitized dataset. Moreover, Figure 6 also represents the lost rule, which is 7→4. During this hiding process, no ghost rules are generated and the number of transactions modified is 766.

Figure 7 represents the number of steps involved in the PPGA. It also depicts the number of repetitions or iterations of the PPGA. In addition, Figure 7 also shows what the support and confidence values of the SARs will be for the next generation. The process will stop when the support or confidence drops below the user-specified threshold. It also indicates that each iteration of the PPGA decreases the confidence of the rule.

To test and validate the PPGA, experiments are conducted on the Zoo dataset [34], Synthetic dataset [35], and Extended Bakery dataset [36]. Moreover, the experiments are performed on those data items that are in Boolean format or are convertible to Boolean format. The data items that we cannot convert to Boolean data will be removed, as shown in Table 10.

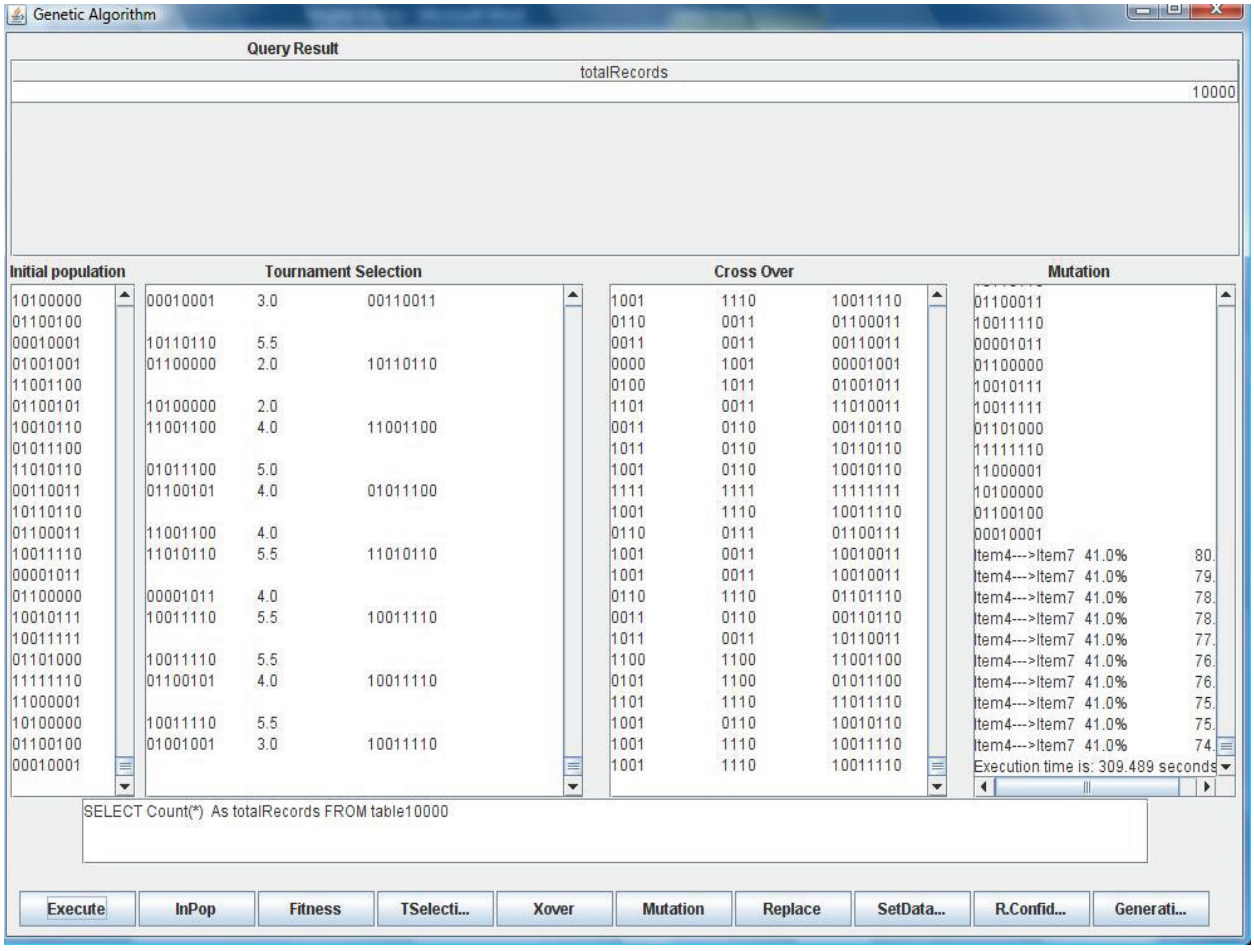

Figure 7. Hiding process of the PPGA.

Table 10. Dataset.

Dataset	Total records	Total attributes	Ordinal attributes
Zoo	101	17	15
Synthetic	10,000	8	8
Extended Bakery	20,000	50	50

4. Results and discussion

In this section, we perform some experiments on each dataset described in the previous section. Initially, the MST is set for each experiment. In the first step, the frequent k-itemset is mined from each dataset. After this, ARs are generated from the frequent k-itemset. Figure 8 depicts frequent k-itemsets (Fk-itemset) and their corresponding ARs with some MSTs. The X-axis represents the sizes of the Zoo, Synthetic, and Extended Bakery datasets, and the left-hand side of the Y-axis describes the MST for each dataset. The right-hand side of the Y-axis indicates the number of frequent k-itemsets and their corresponding ARs generated for each dataset. As the MST decreases the number of Fk-itemsets, the ARs will increase and vice versa, considering that some of these ARs leak confidential information. We call this a SAR or sensitive pattern. The SARs are randomly selected on the basis of their support and confidence.

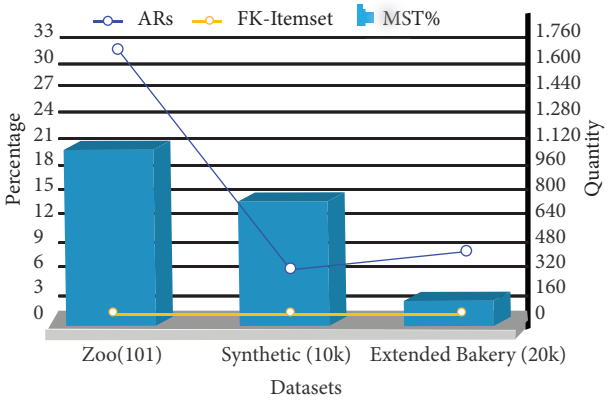

Figure 8. Frequent k-itemsets and their corresponding ARs.

Three parameters play an important role in the rule hiding process, such as the MST, MCT, and number of transactions modified in each generation (TMG) of the PPGA. Therefore, if the values of these parameters are changed, then the result will be changed. Moreover, we conduct several experiments on each dataset. The parameters are set for each experiment. Additionally, the hiding process loses some nonsensitive patterns, called lost rules, and new patterns are also generated, called ghost rules. Thus, the optimal sanitization is a NP-hard problem. The PPGA preserves the privacy of the restrictive patterns by decreasing the ghost rules to 0 and the lost rules to 1 in most of the cases.

Figure 9 illustrates the experimental results of the PPGA. The X-axis describes the sizes of the different datasets and the left-hand side of the Y-axis indicates the different parameter settings and number of transactions modified during the hiding process. Similarly, the right-hand side of the Y-axis represents the lost rule and ghost rule side effects. In other words, the bar represents the deferent parameter settings and modified transactions, while the line represents the number of rules lost and generated due to sanitization. This shows that the proposed technique generates between 0 and 3 lost rules, and minimizes the ghost rule side effects to 0. It also shows the number of transactions modified during the hiding process. The flow of the graph shows that as the

size of the database is increased, the side effects of the lost rules, ghost rules, and transaction modifications are decreased.

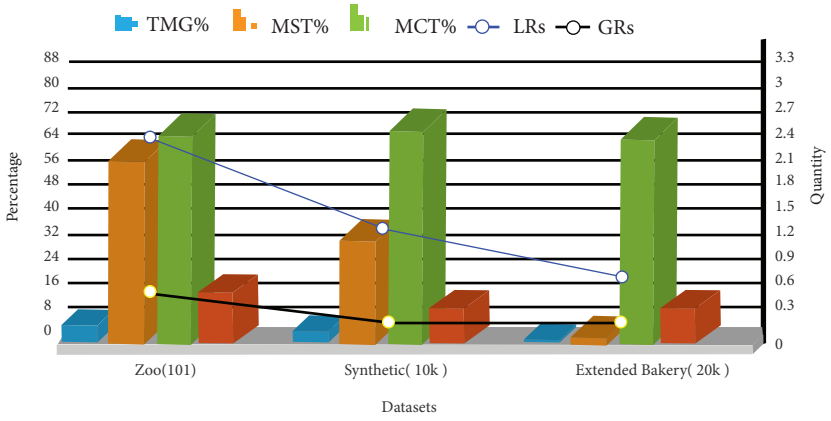

Figure 9. Experimental results of the PPGA.

On the basis of the experimental results, we claim that the proposed architecture hides SARs successfully with no hiding failure. Moreover, the approach used in this work minimizes the side effects of the lost rules and ghost rules. Furthermore, minimizing the number of transaction modifications remains. Additionally, one accidental measure that we find is the CPU time, which is the amount of time taken by the PPGA to preserve the privacy of the confidential information. For small datasets, it is not a problem; however, for large datasets, the PPGA requires a huge amount of CPU time to preserve the privacy of the ARs.

The idea of using a GA for preserving the privacy of SARs was first introduced by Dehkordi et al. [26]. They performed experiments on example datasets that contained 5 transactions and 6 items in each record. They did not perform experiments on large databases. Therefore, we do not compare the proposed technique to their work. The proposed technique is compared to the techniques presented by Verykios et al. [24], Chih-Chia et al. [31], and Naeem et al. [32].

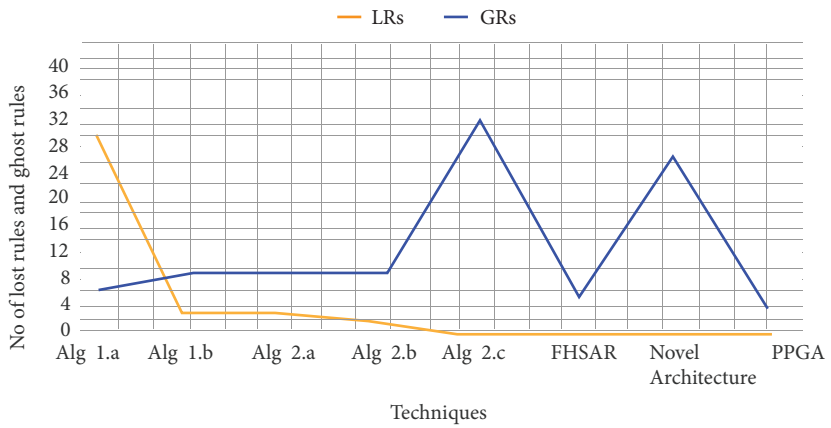

Figure 10. Comparison of the PPGA with the existing techniques averaged over 3 datasets: Zoo, Synthetic, and Extended Bakery.

Figure 10 represents the comparison of the PPGA to other techniques in the literature. It shows that these techniques minimize the side effects in one direction, such as the minimized ghost rule side effects and remaining

or ignored lost rule side effects. It also shows that the proposed technique hides SARs by decreasing the ghost rule side effects to 0. Figure 10 also shows that the PPGA generates between 0 and 3 lost rules. Similarly, the proposed technique hides SARs successfully with no hiding failure. On the basis of such a comparison, the claim is validated that the PPGA outperforms the other techniques presented in the literature.

5. Conclusion

Organizations often share data in order to achieve mutual benefits. However, sharing data most often discloses confidential data. Therefore, data modification or data sanitization techniques are applied to preserve the confidentiality of their confidential data or restrictive patterns in the form of SARs. Moreover, it preserves the privacy of restrictive patterns by concealing the frequent itemsets subsequent to those patterns. This process overcomes the leak of confidential information while sharing data. By doing this, it has an impact on the effectiveness of the data in the form of lost nonrestrictive patterns and new pattern generation. The problem of optimal sanitization is very complex or NP-hard [6]. In the current research work, a GA is used to minimize the impact of data hiding. In the same direction, a new fitness strategy is defined that calculates the fitness of each transaction. Moreover, the proposed technique hides sensitive patterns or SARs successfully, as the technique only modifies those transactions in which sensitive items are present. Furthermore, the technique presented in this paper minimizes the lost rules, as the technique selects those transactions to modify that contain the maximum number of sensitive items and minimum number of availability of nonsensitive items. Similarly, the PPGA generates 0 ghost rules because the selected transactions are replaced by those offspring in which the minimum number of sensitive items are available and the maximum number of nonsensitive items are unavailable. Additionally, to test and validate the PPGA, experiments are performed on the Zoo, Synthetic, and Extended Bakery datasets. Similarly, the experimental results of the PPGA are compared to the techniques presented by Verykios et al. [24], Chih-Chia et al. [31], and Naeem et al. [32]. Thus, the claim is verified that the PPGA outperforms the other techniques in the literature.

In the future, we will design a confidence base PPGA. It will improve the existing fitness function of the PPGA. Moreover, it will modify those items in a sensitive transaction that will reduce the confidence of the rule. Additionally, we shall apply other evolutionary approaches, to preserve the privacy of SARs.

References

[1] R. Agrawal, R. Srikant, "Privacy preserving data mining", ACM SIGMOD International Conference on Management of Data, Vol. 29, pp. 439–450, 2000.

[2] L. Brankovic, V. Estivill-Castro, "Privacy issues in knowledge discovery and data mining", Australian Institute of Computer Ethics Conference, pp. 89–99, 1999.

[3] C. Clifton, D. Marks, "Security and privacy implications of data mining", ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, pp. 15–19, 1996.

[4] Y. Lindell, B. Pinkas, "Privacy preserving data mining", Proceedings of the CRYPTO, pp. 36–54, 2000.

[5] D.E. O'Leary, "Knowledge discovery as a threat to database security", Proceedings of IEEE Knowledge Discovery in Databases, pp. 507–516, 1991.

[6] V. Verykios, E. Bertino, I.G. Fovino, L.P. Provenza, Y. Saygin, and Y. Theodoridis, "State-of-the-art in Privacy Preserving Data Mining", SIGMOD Record, Vol. 33, pp. 50–57, 2004.

[7] D. Agrawal, C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms", Proceedings of the 20th Conference on Principles of Database Systems, pp. 247–255, 2001.

[8] C. Clifton, "Protecting against data mining through samples", Proceedings of the IFIP WG 11.3 13th International Conference on Database Security, pp. 193–207, 1999.

[9] C. Clifton, "Using sample size to limit exposure to data mining", Journal of Computer Security, Vol. 8, pp. 281–307, 2000.

[10] C. Clifton, M. Kantarcioglu, X. Lin, M. Zhu, "Tools for privacy preserving distributed data mining", Proceedings of the SIGKDD Explorations, Vol. 4, pp. 28–34, 2002.

[11] E. Dasseni, V.S. Verykios, A. Elmagarmid, E. Bertino, "Hiding association rules by using confidence and support", Proceedings of 4th Information Hiding Workshop, pp. 369–383, 2001.

[12] S. Oliveira, O. Zaiane, "Privacy preserving frequent itemset mining", Proceedings of the IEEE 14th International Conference on Data Mining, Vol. 14, pp. 43–54, 2002.

[13] S. Oliveira, O. Zaiane, "Algorithms for balancing privacy and knowledge discovery in association rule mining", Proceedings of the 7th International Database Engineering and Applications Symposium, pp. 54–63, 2003.

[14] S. Oliveira, O. Zaiane, "Protecting sensitive knowledge by data sanitization", Proceedings of the IEEE 3rd International Conference on Data Mining, pp. 613–616, 2003.

[15] Y. Saygin, V.S. Verykios, C. Clifton, "Using unknowns to prevent discovery of association rules", SIGMOD Record, Vol. 30, pp. 45–54, 2001.

[16] Evfimievski, R. Srikant, R. Agrawal, J. Gehrke, "Privacy preserving mining of association rules", Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 217–228, 2002.

[17] Evfimievski, "Randomization in privacy preserving data mining", Proceedings of the SIGKDD Explorations, Vol. 4, pp. 43–48, 2002.

[18] Evfimievski, J. Gehrke, R. Srikant, "Limiting privacy breaches in privacy preserving data mining", PODS, pp. 211–222, 2003.

[19] M. Kantarcioglu, C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data", ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 2002.

[20] J. Vaidya, C.W. Clifton. "Privacy preserving association rule mining in vertically partitioned data", Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 639–644, 2002.

[21] R. Agarwal, T. Imielinski, A. Swami, "Mining associations between sets of items in large databases", ACM SIGMOD International Conference on the Management of Data, Vol. 22, pp. 207–216, 1993.

[22] S.L. Wang, A. Jafari, "Using unknowns for hiding sensitive predictive association rules", Proceedings of the IEEE International Conference on Information Reuse and Integration, pp. 223–228, 2005.

[23] C.C. Aggarwal, P.S. Yu, Privacy-Preserving Data Mining: Models and Algorithm, Springer, 2008.

[24] V.S. Verykios, A. Elmagarmid, E. Bertino, Y. Saygin, E. Dasseni, "Association rules hiding", IEEE Transactions on Knowledge and Data Engineering, Vol. 16, pp. 434–447, 2004.

[25] K. Duraiswamy, D. Manjula, N. Maheswari, "A new approach to sensitive rule hiding", Journal of Computer and Information Science, Vol. 1, 2008.

[26] M.N. Dehkordi, K. Badie, A.K. Zadeh, "A novel method for privacy preserving in association rule mining based on genetic algorithms", Journal of Software, Vol. 4, pp. 555–562, 2009.

[27] S.L. Wang, A. Jafari, "Hiding sensitive predictive association rules", IEEE International Conference on Systems, Man and Cybernetics, Vol. 1, pp. 164–169, 2005.

[28] Y. Saygin, V.S. Verykios, A.K. Elmagarmid, "Privacy preserving association rule mining", Proceedings of the 12th International Workshop on Research Issues in Data Engineering, 2002.

[29] Clifton, D. Marks, "Security and privacy implications of data mining", Proceedings of the ACM Workshop Research Issues on Data Mining and Knowledge Discovery, pp. 15–19, 1996.

[30] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, V. Verykios, "Disclosure limitation of sensitive rules", Proceedings of the IEEE Knowledge and Data Engineering Exchange Workshop, pp. 45–52, 1999.

[31] W. Chih-Chia, C. Shan-Tai, L. Hung-Che, "A novel algorithm for completely hiding sensitive association rules", Proceedings of the IEEE 8th International Conference on Intelligent Systems Design and Applications, Vol. 3, pp. 202–208, 2008.

[32] M. Naeem, S. Asghar, "A novel architecture for hiding sensitive association rules", Proceedings of the DMIN, pp. 380–385, 2010.

[33] J. Holland, Genetic Algorithm, Scientific American, 1992.

[34] Frank, A. Asuncion, "{UCI} machine learning repository", University of California, Irvine, School of Information and Computer Sciences, Available at: http://archive.ics.uci.edu/ml, 2010, Last accessed: 02.03.2012.

[35] H. Hamilton, "DBD: data mining projects", University of Regina Available at: http://www2.cs.uregina.ca/∼dbd/cs831/index.html, 2000–9, Last accessed: 15.03.2012.

[36] Track Open Source Project, "Extended bakery dataset", Integrated SCM & Project Management, Available at : https://wiki.csc.calpoly.edu/datasets/wiki/ExtendedBakery20k, 2003, Last accessed: 02.03.2012.