A Database Sanitizing Algorithm for Hiding Sensitive Multi-Level Association Rule Mining

Saad M. Darwish, Magda M. Madbouly, and Mohamed A. El-Hakeem

***Abstract*—The sharing of information has been proven to be beneficial for business partnerships in many application areas such as business planning or marketing. Today, association rule mining imposes threats to data sharing, since it may disclose patterns and various kinds of sensitive knowledge that are difficult to find. Such information must be protected against unauthorized access. The challenge is to protect actionable knowledge for strategic decisions, but at the same time not to lose the great benefit of association rule mining. To address this challenge, a sanitizing process transforms the source database into a released database in which the counterpart cannot extract sensitive rules from it. Unlike existing works that focused on hiding sensitive association rules at a single concept level, this paper emphasizes on building a sanitizing algorithm for hiding association rules at multiple concept levels. Employing multi-level association rule mining may lead to the discovery of more specific and concrete knowledge from datasets. The proposed system uses genetic algorithm as a biogeography-based optimization strategy for modifying multi-level items in database in order to minimize sanitization's side effects such as non-sensitive rules falsely hidden and fake rules falsely generated. The new approach is empirically tested and compared with other sanitizing algorithms depicting considerable improvement in completely hiding any given multi-level rule that in turn can fully support security of database and keeping the utility and certainty of mined multi-level rules at highest level.**

***Index Terms*—Database sanitization, genetic algorithm, privacy preserving data mining, multi-level association rule hiding.**

I. Introduction

In recent years, more and more researches in data mining emphasize the seriousness of the problems about privacy. Privacy issues in data mining cannot simply be addressed by restricting data collection or even by restricting the use of information technology. A key problem faced is the need to balance the confidentiality of the disclosed data with the legitimate users' needs of the data. Privacy preserving data mining (*PPDM*) come up with the idea of protecting sensitive data or knowledge to conserve privacy while data mining techniques can still be applied efficiently [1]. There have been two types of privacy concerning data mining [2], [3]: (1) data privacy, and (2) information privacy. In data privacy, the database is modified in order to protect sensitive data of individuals. Whereas in information privacy (e.g. clustering or association rule), the modification is done to protect sensitive knowledge that can be mined from the database. In other words data privacy is related to input privacy while information privacy is related to output privacy.

In general, classification rules privacy-preserving methods attempt to prevent disclosure of sensitive data so that using non-sensitive data to infer sensitive data becomes more difficult [4]. However, they do not prevent the discovery of the inference rules themselves. Accordingly, scholars have paid attention to the association rules privacy-preserving in recent years. Sensitive association rule hiding is a subfield of *PPDM*, which belongs to output privacy. A sensitive association rule that should be hidden is called a restrictive rule. Restrictive rules always can be generated from frequent itemsets. Therefore, hiding a restrictive itemset implies hiding all the rules which contain the itemset. Such a frequent itemset is called the restrictive itemset. Association rule mechanisms have widely been applied in various businesses and manufacturing companies across many industry sectors such as marketing, forecasting, diagnosis and security [3].

In the literature, various architectures are being examined to design and develop database sanitizing algorithms that make sensitive information in non-production databases safe for wider visibility [4]-[6]. These algorithms can be classified into the following dimensions: (1) algorithms use the support or the confidence of the rule to drive the hiding process; (2) algorithms modify raw data that include the distortion or the blocking of the original values. Data distortion techniques try to hide association rules by decreasing or increasing support. To increase or decrease support, they replace 0's by 1's or vice versa in selected transactions. Data blocking techniques replace the 0's and 1's by unknowns "?" in selected transaction instead of inserting or deleting items; (3) algorithms hide sensitive rules by setting their confidence below a user-specified threshold or sensitive items by hiding the frequent itemsets from which they are derived; and (4) algorithms hide single rule or multiple rules during an iteration.

Regarding the nature of the hiding algorithm, sanitizing algorithms can be categorized into: heuristic, border, exact, or reconstruction (reform) based algorithms [3]-[5], [7], [8]. Heuristic approaches use trials for modifications in the database. These techniques are efficient, scalable and fast, however they do not give optimal solution and also are *CPU*-intensive and require various scans depending on the number of association rules to be hidden. Border based approaches track the border of the non-sensitive frequent item sets and greedily apply data modification that may have minimal impact on the quality of the border to accommodate the hiding sensitive rules. These approaches outperform the heuristic one and causing substantially less distortion to the original database to facilitate the hiding of the sensitive knowledge. Yet, in many cases these approaches are unable

Manuscript received January 5, 2014; revised April 3, 2014.

All authors are with the Institute of Graduate Studies and Research, Alexandria University, 163Horreya Avenue, El-Shatby, 21526 P.O. Box 832, Alexandria, Egypt (e-mail: Saad.darwish@alex-igsr.edu.eg, m.madbouly@alex-igsr.edu.eg, m_abdelhakeem@ymail.com).

DOI: 10.7763/IJCCE.2014.V3.337

to identify optimal hiding solutions, although such solutions may exist for the problem at hand.

Exact approaches are considered as non-heuristic algorithms which envisage the hiding process as a constraint satisfaction problem that may be solved using linear programming. These approaches provide better solution than other approaches and can provide optimal hiding solution with ideally no side effects, but they suffer from high degree of difficulty and complexity. Finally, reconstruction based approaches conceal the sensitive rules by sanitizing itemset lattice rather than sanitizing original dataset. Compared with original dataset, itemset lattice is a medium production that is closer to association rules. These types of approaches generate lesser side effects in database than heuristic approaches. Despite its benefits, sanitization of the new database from scratch becomes impractical and this should be avoided. Table I offers a comparative view of the previous approaches. Readers looking for more information regarding these approaches can refer to [9].

TABLE I: A COMPARISON TABLE [7]

Approaches	Execution time	Scalability	Hiding Failure	Information Loss	Modification Degree
Heuristic	Fast	Good	Very low	Moderate	Moderate
Border	Moderate	Moderate	None	Good	Good
Exact	Slow	Low	None	None	Very good
Reform	Slow	Low	None	Good	Moderate

At recent, many works have been focused on mining association rules at a single concept level [2], [7], [10], [11]. There are applications which need to find associations at multiple concept levels. Multi-level association rules, first introduced in [12], use hierarchy concept defined as relations of type 'is-a' between objects to extract rules that items belong to different levels of abstraction. For example, “people who buy computer also buy printer”. In this example, computer and printer each contains a hierarchy of different types and brands. To explore multi-level association rule mining, one needs to provide data at multi-levels of abstraction and to own efficient methods for multi-level rule mining. The first requirement can be satisfied by providing concept taxonomies from the primitive level concepts to higher levels [13].

Mining knowledge at multi-levels may help database users find some interesting rules which are difficult to be discovered otherwise and view database contents at different abstraction levels and from different angles. Furthermore, multi-level rules can provide richer information than single level rules, and represent the hierarchical nature of the knowledge discovery process. Rules regarding item sets at suitable levels could be relatively functional. It can help organizations to make promotional strategies and help enhancing the sales and setting the future plans [13], [14].

For analyzing the performance of any sanitizing algorithm the researchers have considered the following factors [8], [15] that are considered as side-effects of the modification process. (1) Hiding failure : the portion of sensitive rules that are not hidden after applying the sensitive-rule-hiding procedure; (2) False rules: can be quantified as the number of ghost rules in the sanitized database; (3) lost rules: can be calculated as the number of non-sensitive rules that become infrequent in the sanitized database; (4) Execution time: the time needed to execute the algorithm; (5) Modification degree: can be measured as the difference between the original and sanitized databases. Robust sanitizing algorithm must minimize the previous side effects. In general, the correlation among rules can make it impossible to achieve this goal.

The main contribution of this paper is to propose a heuristic based sanitizing algorithm for hiding sensitive multi-level association rules. The proposed algorithm utilizes genetic algorithm as an optimization technique for selecting itemsets to be sanitized (changed) from transactions that support sensitive rules with the aim of making minimum modification in original database. Although there were many studies using heuristic approach, they all focus on hiding association rules at a single concept level. Nevertheless, undesired side effects, e.g., non-sensitive rules falsely hidden and spurious rules falsely generated may be produced in the rule hiding process. The proposed algorithm is efficient, fast, and gives optimal solution for fading the side effects.

The structure of the paper is as follows: Next section gives a short survey about association rule hiding algorithms. In Section III, our algorithm to protect sensitive multi–level rules in association rule mining is explained. The experimental results that present the performance and various side effects of the proposed algorithm are given in Section IV. Then the paper is concluded with our final remarks on the study and the future work in Section V.

II. SOME RELATED EARLIER WORKS

In the terrain of privacy preserving data mining many studies have been carried out for protecting sensitive association rules in database. A good number of algorithms are reported in the literature for heuristic –based single level association rule hiding, which has been developed using techniques from mathematics, statistics, and computer science. For example, authors in [16] proposed a heuristic algorithm that relies on Boolean association rules; aiming at selectively hiding some frequent itemsets from large databases with as little impact on other non-sensitive frequent itemsets as possible. Specifically, the authors dealt with the problem of modifying a given database so that the support of a given set of sensitive rules decreases below the minimum support value.

Y. Wu *et al*. [17] suggested a heuristic method that could hide sensitive association rules with limited side effects. They remove the disjoint assumption (the sensitive frequent itemsets appearing in a sensitive rule do not appear in any other sensitive rule) and allow the user to select sensitive rules from all strong rules. In their algorithm, item conflict degree helps to minimize the non-sensitive patterns lost during sanitization. When the size of database is large, the time consuming of their algorithm will smaller than traditional sanitizing algorithms. Differentiate from the previous distortion-based algorithms Saygin *et al*. [18] described blocking concept to prevent the discovery of sensitive rules, which applies unknown values "?" to replace original values. Their work presented the concept of fuzzification of the support and the confidence metrics. However, the side effects will be out of control since they do not consider the correlation among rules in their modification scheme. Another related work presented in [19] where the authors proposed a rule hiding algorithm that correlates sensitive association rules and transactions by using a graph to effectively select the proper item for modification. The algorithm can completely hide any given sensitive association rule by scanning database only once, which significantly reduces the execution time.

Unlike previous methods that dealing with hiding one rule at a time, multiple rules hiding approach was first introduced in [20]. Four heuristic algorithms were proposed that select the sensitive transactions to sanitize based on degree of conflict and then remove items from selected transactions based on certain criteria. Their proposed algorithms are efficient and require two scans of the database, regardless of the number of sensitive item sets to hide. In this case, a transaction retrieval engine is used to speed up the process of finding the sensitive transactions that are identified according to the sensitive patterns. How to choose the sensitive transactions and how to choose the victim items from the sensitive transactions are the two most important issues in it.

In the same direction, and to enhance the multiple rule hiding algorithms, the authors in [21] presented a sliding window algorithm (*SWA*) to scan a group of transactions at a time. This algorithm is useful for sanitizing large transactional databases based on a disclosure threshold (or a set of thresholds) controlled by a database owner. A strong point of *SWA* is that it does not introduce false drops to the data. In addition, *SWA* has the lowest misses cost among the known existing sanitizing algorithms. A short summary of the existing literature on single level association rule hiding algorithms can be found in [7], [9].

In [22] the idea of using correlation matrix for hiding sensitive patterns is introduced. The authors proposed three multiple association rule hiding heuristics data distortion approaches that operate on a sanitization matrix and then multiply with original database to obtain a sanitized database. Instead of selecting individual transactions and sanitizing them, the authors proposed a methodology for directly constructing a sanitization matrix by observing the relationship that holds between sensitive patterns and non-sensitive ones.

Soft computing especially genetic algorithms seems to be an appropriate paradigm for hiding the sensitive rules in the heuristic algorithms only in the case that an optimal solution does not exist. There are many mechanisms that adapt genetic algorithm for hiding single level association rules. In [23] the authors explored new multi-objective method for hiding sensitive association rules based on the concept of genetic algorithms. They have used four fitness strategies that rely on minimizing number of sensitive rules and maximizing number of non-sensitive association rules that can be extracted from sanitized dataset. Similarly S. Narmadha et al. [24] investigated how sensitive rules in one level concept should be protected from malicious data miner and proposed genetic algorithm technique for hiding the sensitive rules. In genetic algorithm, a new fitness function is calculated, based on this value the transactions are selected and the sensitive items of this transactions are modified with crossover and mutation operations without any loss of data. In their technique, all the sensitive rules are hidden, no false rules can be generated, and non-sensitive rules are not affected.

Pinning our attention to the work done by R. A. Shah *et al*. [25] in which a new modification technique called privacy preserving genetic algorithm is introduced. This technique modifies the database recursively until the support or confidence of the restrictive patterns drop below the user specified threshold. The technique is only applicable on binary dataset. In addition, the technique only modifies those transactions which contain maximum number of sensitive items and minimum number of availability of non-sensitive items.

Regarding multi-level association rule hiding, only the work suggested in [26] is found in the literature. The authors applied sensitive itemsets hiding algorithm through the insertion of a minimal extension to the original database (i.e. additive model for sensitive item set hiding). In their extended algorithm, the size of the additional transactions to be added is calculated based on obtained minimum support and original database minimum support. The database is updated with the new extended database which hides frequent sensitive itemsets. Their proposed methodology is capable of identifying an ideal solution whenever one exists, or approximate the exact solution.

Following this recent development, this paper presents a novel approach for hiding multi-level association rules. The work recommended in this paper try to remedy the limitations of the algorithm presented in [25], which includes increasing the size of the databases and minimizing the availability of database through hiding certain itemsets instead of rules. To the best of our knowledge, apart from ongoing research work regarding a distortion model for sensitive rule set hiding; the proposed system facilitates rule hiding in multi-level databases without extending the original database (no dummy transactions) and with minimum lost and ghost rules.

III. Our Approach for Multilevel Rule Hiding

Fig. 1 shows the proposed database sanitizing algorithm that consists of four major steps: (1) Build encoded transaction table, (2) transform the transaction dataset to Boolean form, (3) Generate multiple-level association rules, (4) Items selection for modification using genetic algorithm. Unlike previous database sanitizing efforts based on multi-level association rule in which specific items are hidden instead of specific rules; the proposed sanitizing algorithm employs genetic algorithm to select the best items for modification to hide sensitive multi-level association rules. Note that hiding itemset prevents itemsets from being appeared in any rules exceeding minimum confidence whether those rules are sensitive or non-sensitive, but hiding certain rules tries to modify the itemsets contained in these rules to only reduce the confidence of the sensitive rules below a user-specified threshold to hide them and will make the same items contained in sensitive rules free to appear in other non-sensitive rules which will finally lead to more data availability for users' needs [13].

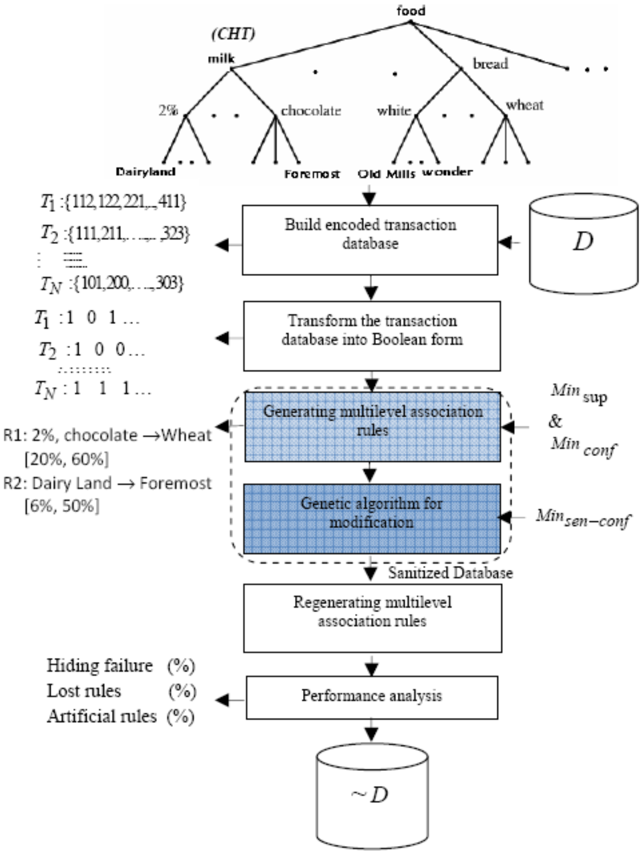
food
(CHT)
milk
bread
2%
chocolate
white
wheat
Dairyland
Foremost
Old Mills
wonder
T_1 :{112,122,221,..,411}
T_2 :{111,211,..., 323}
T_N :{101,200,...,303}
Build encoded transaction database
D
T_1 :1 0 1 ...
T_2 :1 0 0 ...
T_N : 1 1 1 ...
Transform the transaction database into Boolean form
$Min_{\sup}$ & Min_{conf}
Generating multilevel association rules
R1: 2%, chocolate → Wheat [20%, 60%]
R2: Dairy Land → Foremost [6%, 50%]
Genetic algorithm for modification
$Min_{sen-conf}$
Sanitized Database
Regenerating multilevel association rules
Hiding failure (%)
Lost rules (%)
Artificial rules (%)
Performance analysis
~D

Fig. 1. Proposed sanitizing system architecture.

The problem discussed in this paper can be formulated as follows: Consider a given database *D* which provides data at multi-levels of abstraction called Concept Hierarchy Tree (CHT), minimum support threshold value $(Min_{\sup})$, minimum confidence threshold value (Min_{conf}), for each level of abstraction, a set of multi-level association rule MAR that can be mined from *D* and a set of sensitive multi-level association rules $(MAR_{sen} \subseteq MAR)$ to be hidden then to generate a novel database ~*D* with goals of (1) No MAR_{sen} should be revealed, (2) All the multi-level non-sensitive rules $(MAR_{non-sen} = MAR - MAR_{sen})$ can be successfully mined in the sanitized database ~*D* and (3) No rule that was not found in the original database *D* can be found at the sanitized database ~*D* under the same threshold values $(Min_{\sup})$ and (Min_{conf}) (or at any value higher than these thresholds). In our case, utilizing genetic algorithm for items selection improves the system ability to make little modifications in the original *D* to achieve best rates of the previous goals. The following steps are required for the proposed solution:

Step 1: Input a database with multi-level concept hierarchy: Here, we consider that the database contains: (1) A transaction data set *T* which consists of a set of transactions $\langle T_r, \{A_p, \ldots\ldots A_q\}\rangle$ where T_r is a transaction identifier, $A_i \in I$ (for $i = p, \ldots, q$), and I is the set of all the data items in the item data set; and (2) the description of the item data set, which contains the description of each item in I in the form of $\langle A_i, descripton\rangle$ as illustrated in Table II to Table IV[12].

TABLE II: A SALES TRANSACTION TABLE

Transaction_id	Bar_code_set
351428	{17325, 92108, 55349,….}
982510	{92458, 77451, 60395,….}

TABLE III: A SALES_ ITEM (DESCRIPTION) TABLE

Bar-code	Category	Brand	Content	Size	price	…
17325	Milk	foremost	2%	1(ga.)	$3.89	
…	…	…	…	…	…	…

TABLE IV: A GENERALIZED SALES_ ITEM DESCRIPTION TABLE

GID	Bar_code_set	Category	Content	Brand
112	{17325,3141, 91265}	Milk	2%	foremost
…	…	…	…	…

Based on the item description, CHT is built. CHT is modeled by a directed acyclic graph as shown in Fig. 1. An arc of CHT represents an "is-a" relationship between the source and the destination. Transactions I contain only the items belonging to the lowest level (Terminal level). In taxonomy, levels are numbered from 0, as the level 0 represents the level root. Items belonging to a level l are numbered with respect to their parent in an ascending order. In many applications, concept hierarchies may be specified by users familiar with the data, or may exist implicitly in the data. In our case, we assume that the taxonomy information is provided implicitly in dataset [15].

Step 2: Build encoded transaction table: The actual data converted into a hierarchy-information encoded transaction table. The encoding refers to the process of specifying node id to each item in the concept hierarchy of items in such that the *id* self-contain taxonomy information about the concept hierarchy. The transaction table represents the data where each instance in the dataset represents one transaction in the form of $\langle T_r \; id \text{ of } I \rangle$ (see Fig. 1). Herein, an encoded string, which represents a position in a hierarchy, requires fewer bits than the corresponding object identifier or bar-code. Moreover, encoding makes more items to be merged (or removed) due to their identical encoding, which further reduces the size of the encoded transaction table.

Step 3: Transform the transaction database into Boolean form: We set up a Boolean matrix A_{r*n}, which has *r* rows and *n* columns. Scanning the transaction database *D*, if item I_i is in transaction T_r, where $1 \le i \le n$, the element value of I_i is '1,' otherwise the value of I_i is '0'. This stage simplifies the processing of next steps.

Step 4: Generating the Multi-level association rules: The proposed system uses the same theory introduced in [12] for multiple-level association rules construction. Formally, given Encoded transaction table (*ET*), Min_{sup}, Min_{conf}, threshold for each level *L*, the procedure for progressive and deepening approach is as follow:

```
For each level L
  $Cand_L$ ← The candidate large 1- itemsets (descendants of the previous level large 1- itemsets)
  Scan ET and remove those candidates in $Cand_L$ with
  $Support_L < Min_{\sup,L}$
  $k = L \leftarrow 2$
    Loop
      $Cand_k \leftarrow$ Generate the candidates from the ($k$-1) itemsets
      For each transaction $T_r$ in encoded table
        1- Increment the support for each candidate k itemsets that appears in $T_r$
        2- Remove those members of $Cand_k$ who have support less than $Min_{\sup}$
        3- If $Cand_k$ is empty then break from loop
      End for
    End loop
  LargeSets$_i$ $A_i \leftarrow$ The union of all non-empties $Cand_k$
End for
Return the union of all LargeSets$_i$
For each large item set $A_i$
  For every proper subset $B$ of $A$
    If $(\sigma(A)/\sigma(A-B > Min_{conf}))$
      Append $(A-B)$ to Valid Rule Set $MAR$
  End
End
```

This progressive and deepening (level 1, level 2, level 3, etc.) approach continues at every lower level and incrementally within each level until no large frequent-itemsets can be found. In this case $Min_{\sup}$ and Min_{conf} ,extracted with the help of equation 1, and 2 respectively, varies from level to level i.e. both are reduced going from higher to lower levels by using δ operator (defined by the owner)[25].

$$Support = \varphi(I_{lh} \rightarrow I_{rh}, T) = \frac{|I_{lh} \cup I_{rh}|}{|N|} \ge Min_{\sup} \quad (1)$$

$$Confidence = \varphi(I_{lh} \rightarrow I_{rh}, T) = \frac{|I_{lh} \cup I_{rh}|}{|I_{lh}|} \ge Min_{conf} \quad (2)$$

$I_{lh}, I_{rh} \in I$ are the left hand side and the right hand side itemsets of each multi-level rule respectively and *N* is the number of transactions in *D.* For more comprehensive details readers can refer to [13], [14]. Based on the discovered multi-level rules and privacy requirements, hidden multi-level rules or patterns (MAR_{sen}) are then selected depending on $Min_{sen-conf}$, threshold satisfying that $Min_{sen-conf} > Min_{conf}$, ($Min_{sen-conf} > 70\%$ in our case). This threshold indirectly controls the proportion of transactions to be sanitized.

Step 5: Genetic algorithm for modification: This step represents the main contribution of the proposed system for hiding sensitive multi-level association rule. Given the sensitive rules from the previous step, the system tries to hide these rules by utilizing the procedure of reducing their confidence below Min_{conf} by means of increasing support of the antecedent and decreasing support of the consequent via replacing 1's by 0's for items and vice versa in the transactions. Since the modification of all sensitive itemsets associated with sensitive rules in all database's transactions will make the algorithm *CPU*-intensive.

In current research work, our solution to tackle the above problem is via employing Genetic Algorithm (*GA*) to select best itemsets for modification. Therefore there is no need to modify all of the transactions in our algorithms. With this step we can reach to better performance of sanitization speed and less number of modification needed in hiding process. Furthermore, the technique can be applicable for small dataset as well as for large dataset.

GA allows a population composed of many individuals to develop under particular selection rules to a state that maximizes the "fitness" (i.e., minimizes the cost function). The proposed system utilizes two versions of fitness function that only modifies those transactions which contain maximum number of sensitive items and minimum number of availability of non-sensitive items with the aim of minimizing both lost and ghost rules. In both of them, the transaction having lower fitness value will be selected for modification. The first fitness function is defined as [24]:

$$\forall f_v \in T_r \;\; f_{v1} = \frac{X_r + Y_r}{2} \tag{3}$$

$$X_r = \sum_{i=1}^{n}(I_i = 1), \;\; Y_r = (S_r \text{ in } T_r) \tag{4}$$

where $S_r \in I$, denotes a set of sensitive items, $I \in T$, $T_r\{T_1, T_2, ..., T_N\}$ symbols transaction, *n* represents the number of items in each transaction, *r* characterizes transaction's number, and *v* is a set of identifier for elements of f, $f_v = \{f_1, f_2, ..., f_N\}$. This version of fitness function relies on item's restriction strategy (i.e. replacing 1's by 0's). Whereas the second function is designed based on weighted sum function and calculated as [26]:

$$W_1 * C_1 + W_2 * (\frac{1}{C_2}) \tag{5}$$

$$\forall C_1 \in T_r, C_1 = 1/\sum_{i=1}^{n} Count(S_r) \text{ in } T_r + \sum_{i=1}^{n}(I_i = 1) \tag{6}$$

$$\forall C_2 \in T_r, C_2 = 1/\sum_{i=1}^{n} Count(S_r) \text{ in } T_r + \sum_{i=1}^{n}(I_i = 0) \tag{7}$$

$$w_1 + w_2 = 1 \text{ (In our case } w_1 = w_2 = \frac{1}{2} \text{)} \tag{8}$$

This version of fitness function relies on item's distortion strategy (i.e. replacing 1's by 0's and vice versa). Equation (6) guarantees that the lost rules are minimized because, the system select those transactions to modify in which less number of data items are available, also Equation (7) insures that ghost rules are minimized because the selected transactions are replaced to those offspring in which maximum number of data items are unavailable.

GA works as follows: First each transaction, related to sensitive items for each sensitive rule, is represented as a chromosome. For the initial population all related transactions are chosen. Based on the survival fitness described above, the population will transform into the future generation through chromosomes selection, crossover, and mutation operations. Selection embodies the principle of "survival of the fittest". Satisfied fitness chromosomes are selected for reproduction. Poor chromosomes or lower fitness chromosomes may be selected a few or not at all. In our case, tournament selection, in which two chromosomes are selected randomly from population and more fit of these two is selected for mating pool, is used. Readers looking for more information regarding using of *GA* for association rules hiding can referee to [23]-[25].

It is known that database transactions contain the items going at lowest level, and the changes are made in those items. If the sensitive items belong to any level other than lowest one, then all the descendants of these items are all sensitive. So, hiding any upper level rules requires knowing descendants of the items contained in this rule and consequently, knowing the transactions associated with these items. *GA* procedure for modifying the sensitive items is as follows:

```
Input: $MAR_{sen}$, $Min_{sen-conf}$, Crossover & Mutation rates,
       No. of generation (g).
Output: Sanitized database $\sim D$.
While $MAR_{sen}$ {}! = Ø OR generation! = g
    For each rule $R \in MAR_{sen}$
      1. Determine CHT's lowest level sensitive itemsets
         $I_{sen} \in R$.
      2. Determine a set $T_{sen} \subseteq T$, where $I_{sen} \in T$.
        For each $T_r \subseteq T_{sen}$
          2.1 Fitness: $f_{v1} = \frac{X_r + Y_r}{2}$
                    OR
              $f_{v2} = W_1 * C_1 + W_2 * (1/C_2)$
          2.2 Selection: Based on the version of fitness
                    Function $f_{v1}$ or $f_{v2}$
          2.3 Crossover: $T_r * T_{r+1}$
          2.4 Mutation:
              (Restriction mode)
              Select $T_r$, Change 1 to 0 in case of $f_{v1}$
                    OR
              (Distortion mode)
              Select $T_r$, Change 1 to 0 or 0 to 1 randomly in
              case of $f_{v2}$
        End for
    End for
End While
```

IV. Experimental Result

The experiments were carried out to show the effectiveness of the proposed system. For the evaluation a database containing 5000 transactions has been used. The database itself consists of one relation (Table) with 50 items (columns) in each record that consists of one identifier's attribute, and forty nine quantitative attributes. Each transaction contains the IDs (items) for products that were purchased by a customer. The experiments are performed on the MySQL 5.2 CE DBMS on Microsoft Windows 7 Enterprise SP1 32 bit running on a machine has the following configurations: Intel Core Duo CPU T2350 @ 1.86 GHz 1.87 GHz, GB of RAM and programmed in the MATLAB language (version 2 7.01) and java (Net Beans IDE 7.3.1).

The side effects of the proposed genetic based multi-level rule hiding approach are evaluated through measuring Hiding Failure (*HF*), Artificial Rule generation (*AR*), and Lost Rules (*LR*), which are defined as follows [8], [15]:

$$HF = \frac{|MAR_{sen}(\sim D)|}{|MAR_{sen}(D)|} \tag{9}$$

$$AR = \frac{|MAR(\sim D)| - |MAR(D) \cap MAR(\sim D)|}{|MAR(\sim D)|} \tag{10}$$

$$LR = \frac{|MAR_{non-sen}(D) - MAR_{non-sen}(\sim D)|}{|MAR_{non-sen}(D)|} \tag{11}$$

where $|.|$ is the size of set. During evaluation, databases with different sizes are generated for the series of experiments from the original database. The average length of transactions' items of each database is 10, 20, and 50 items in the generated databases. The experimental results are obtained by averaging from 5 independent trials with different sanitization factors. Three parameters play an important role in rule hiding process, which are $Min_{sen-conf}$, number of transactions and the number of items. Therefore, if the values of these parameters are changed then the result will be changed. We conducted several experiments on each database to show the influence of these parameters on the suggested system. The specifications of used genetic algorithm for privacy preserving in association rule mining is as follows: Population size varies with the number of transactions, Mutation Rate = 0.01, Crossover Probability = 0.80, Chromosome Length fluctuates with the number of items and Number of Generations =50.

The first experiment finds the relationship between number of hidden multi-level sensitive rules, artificial rules, and lost rules with number of transactions. In this experiment $Min_{\sup} = 25\%$ and $Min_{conf} = 58\%$. $Min_{sen-conf}$ value is taken as 60%, 70%, and 80% for 500, 1000, 2000, 3500 and 5000 transactions respectively. Evaluation of side effects as a result of the hiding process is shown in Table V and VI for fitness functions f_{v1} (restrictive mode) and f_{v2} (distortion mode) respectively. As clarified in both tables, the number of non-sensitive rules lost is quite low and tends to increase when the number of transactions in the database increases and tends to decrease when number of sensitive rules $|R_S|$ decreases. It is found that the hiding failure with the proposed algorithm is zero which means all the sensitive rules are protected from the disclosure. The accuracy of sensitive rule protection is 100%.

As shown in Table VI (distortion mode modification), the number of new rules introduced tends to increase when the number of transactions in the database increases. We experienced that if we hide larger sets of rules, a larger number of new frequent itemsets is introduced and therefore an increasing number of new rules are generated. Unlike the restriction mode modification in which the number of new rules mined from the database after hiding the rules is zero for all database sizes. In the other words, utilizing f_{V1} achieves superior performance in minimizing ghost rules. However, in both cases, we can find that the number of transactions to be modified is smallest, because the suggested system selects transactions which satisfy maximum modification rules' characteristics to modify each time, so it needs much less transaction to be modified overall.

The second experiment compares between our algorithm and the work presented in [26] that deals also with multi-level association rule hiding but from the perspective of hide itemsets instead of rules as the proposed system works. Table VII shows average side effect and CPU-time produced by both systems under the same database with 5000 transactions covering 50 items. The Table shows that only few lost rules were missed by the proposed system. Furthermore, both algorithms produced no ghost rules when hiding the selected rules without any side effects of hiding failure. In summary, the illustrations show that the proposed algorithm outperforms other method in minimizing the side effects, computational complexity, and data distortions. Accordingly, our algorithm causes negligible impact on the quality of the data mining results and required little time when completely hiding many sensitive association rules from the real database.

TABLE V: PERFORMANCE EVALUATION FOR f_{V1}

$Min_{sen-conf}$	60 %			70 %			80 %		
No. of Transactions	*LR* %	*HF* %	*AR* %	*LR* %	*HF* %	*AR* %	*LR* %	*HF* %	*AR* %
1000	0	0	1.28	0	0	1.04	0	0	0.89
2000	0	0	1.42	0	0	1.25	0	0	1.00
3500	0	0	1.70	0	0	1.48	0	0	1.43
5000	0	0	2.13	0	0	2.08	0	0	2.00

TABLE VII: COMPARATIVE RESULTS

factor Algorithm	*LR* (%)	*AR* (%)	*HF* (%)	Accuracy (%)	CPU-time(s)
Proposed	2.13	0	0	100	5
Itemsets-based[26]	7.60	0.16	0	100	13

TABLE VI: PERFORMANCE EVALUATION FOR f_{V2}

$Min_{sen-conf}$	60 %			70 %			80 %		
No. of Transactions	*LR* %	*HF* %	*AR* %	*LR* %	*HF* %	*AR* %	*LR* %	*HF* %	*AR* %
1000	0	0.010	1.30	0	0.008	1.05	0	0.005	0.93
2000	0	0.015	1.44	0	0.012	1.29	0	0.010	1.05
3500	0	0.027	1.71	0	0.017	1.59	0	0.014	1.49
5000	0	0.030	2.15	0	0.020	2.13	0	0.018	2.07

We assessed the time of the proposed system by comparing the execution time required by the algorithm under varied factors such as database size $|D|$ and number of $|R_S|$ concluded through $Min_{sen-conf}$ threshold. The processing time reported includes the *CPU* time consumed in the processing steps (after multi-level sensitive rules have been extracted). We exclude the *I/O* time spent on the index construction and the database modification in order to highlight the impact of the database scale on our modification mechanisms for rule hiding. This comparison is plotted in Fig. 2. As can be seen from the Figure, the time requirements for hiding a set of rules increase linearly with the database size for large data sets as well. In fact the linear behaviour is more obvious for larger scale data. Another observation is that the time requirements for hiding with $Min_{sen-conf} = 60$ are higher than hiding $Min_{sen-conf} = 80$, which is also another expected result (i.e. is scalable with the size of a specified set of sensitive association rules). In average, the proposed system requires only 5 seconds for running 5000 transactions of 50 items (sanitization only). The system is suitable for application in a real business world context.

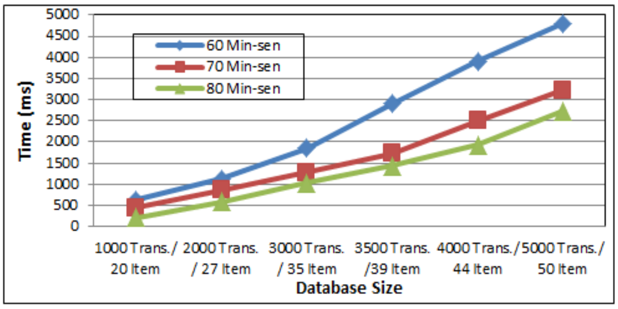

Fig. 2. CPU time at different factors.

Regarding the complexity of the proposed system, the complexity relies on both number of transaction, number of items and complexity of genetic algorithm, which in turn depends on fitness function. The simplest case-roulette wheel selection, point mutation, and one point crossover with both individuals and populations represented by fixed length vectors has time complexity of:

$$O(g*(mutation+crossover+selection)) \qquad (12)$$

where g is the number of generations, *mutation* is the complexity of point mutation ($n_p * m$ with n_p the size of the population and m the size of the individuals), cross the time complexity of *crossover* ($n_p * m$ again), and select the time complexity of *selection* (n_p in the case of an efficiently done roulette wheel). Therefore, the time complexity of a simple genetic algorithm is $O(g*n_p*m)$ as this is the dominating term. So if $|D|=N$ then the total complexity of the algorithm is $O(N*g*n_p*m)$. In summary, the experimental results showed that the proposed algorithm, achieved minimum side effects and *CPU*-Time in the context of hiding a specified set of sensitive multi-level association rules.

V. Conclusion

In this work, the database privacy problems caused by data mining technology are discussed. We have taken heuristic approach based on both distortion and restriction procedures for hiding sensitive multi-level association rules by using genetic optimization algorithm. The proposed approach is based on the strategy to simultaneously decrease the confidence of the sensitive rules. The approach applies minimum number of changes to the database and minimal amount of non-sensitive association rules are missed which is the ultimate aim of data sanitization.

Main strengths of the advised algorithm are (1) this algorithm is useful for sanitizing large transactional databases based on a $Min_{sen-conf}$ threshold controlled by a database owner, (2) simple heuristic method that are used in transaction and item selection for sanitization eliminates the need of extra computational cost, (3) efficiency is increased since victim's items selection is adjusted by using genetic algorithm, (4) data availability is augmented by hiding specific rules instead of items.

Performance evaluation study is done on different databases to show the efficiency of the versions of the fitness function while the size of the original database, the number of itemsets and the $Min_{sen\text{-}conf}$ value change. Future work has to be carried out to develop an optimal database sanitizing algorithm for multi-cross level association rules. Moreover, further research is in progress to develop new fitness functions and applying other optimization techniques to minimize the iterations.

References

[1] M. Patel, A. Hasan, and S. Kumar, "A survey: Preventing discovering association rules for large data base," *International Journal of Scientific Research in Computer Science and Engineering*, vol. 1, issue 3, pp. 35-38, Jun. 2013.

[2] S. Gacem, D. Mokeddem, and H. Belbachir, "Privacy preserving data mining: Case of association rules," *International Journal of Computer Science Issues*, vol. 10, issue 3, no. 1, pp. 91-96, May 2013.

[3] K. Sathiyapriya and G. S. Sadasivam, "A survey on privacy preserving association rule mining," *International Journal of Data Mining & Knowledge Management Process*, vol. 3, no. 2, pp. 119-131, Mar. 2013.

[4] B. Suma, "Association rule hiding methodologies: A survey," *International Journal of Engineering Research & Technology*, vol. 2, issue 6, pp. 181-185, Jun. 2013.

[5] V. S. Verykios, "Association rule hiding methods," *Data Mining and Knowledge Discovery*, vol. 3, issue 1, pp. 28-36, Jan. - Feb. 2013.

[6] K. Shah, A. Thakkar, and A. Ganatra, "A study on association rule hiding approaches," *International Journal of Engineering and Advanced Technology*, vol. 1, issue 3, pp. 72-76, Feb. 2012.

[7] G. Lee and Y. C. Chen, "Protecting sensitive knowledge in association patterns mining," *Data Mining and Knowledge Discovery*, vol. 2, issue 1, pp. 60- 68, Jan.-Feb. 2012.

[8] E. Bertino and I. N. Fovino, "A framework for evaluating privacy preserving data mining algorithms," *Data Mining and Knowledge Discovery*, vol. 11, no. 2, pp. 121-154, Sep. 2005.

[9] A. Tomar, V. Richhariya, and R. K. Pandey, "A comprehensive survey of privacy preserving algorithm of association rule mining in centralized database," *International Journal of Computer Applications*, vol. 16, no. 5, pp. 23-27, Feb. 2011.

[10] K. Shah, A. Thakkar, and A. Ganatra, "Association rule hiding by heuristic approach to reduce side effects & hide multiple R.H.S. items," *International Journal of Computer Applications*, vol. 45, no. 1, pp. 1-7, Nov. 2012.

[11] D. Jain, A. sinhal, N. Gupta, P. Narwariya, D. Saraswat, and A. Pandey, "Hiding sensitive association rules without altering the support of sensitive item(s)," *International Journal of Artificial Intelligence & Applications*, vol. 3, no. 2, pp. 75-84, Mar. 2012.

[12] J. Han and Y. Fu, "Discovery of multiple-level association rules from large databases," in *Proc. Int. Conf. Very Large Data Bases*, Switzerland, Sept. 1995, pp. 420-431.

[13] F. A. El-Mouadib and A. O. El-Majressi, "A study of multilevel association rule mining," in *Proc. the Int. Arab Conf. Information Technology*, Libya, Dec. 2010, pp. 14-16.

[14] S. Bhasgi and P. Kulkarni, "Multilevel association rule based data mining," *International Journal of Advances in Computing and Information Researches*, vol. 1, no. 2, pp. 39-42, Apr. 2012.

[15] E. Bertino, D. Lin, and W. Jiang, *Privacy-Preserving Data Mining: Models and Algorithms*, New York: Springer-Verlag, 2008. ch. 8, pp. 183-205.

[16] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. S. Verykios, "Disclosure limitation of sensitive rules," in *Proc. IEEE Knowledge and Data Engineering Exchange Workshop*, USA, Nov. 1999, pp. 45–52.

[17] Y. Wu, C. Chiang, and L. P. Chen, "Hiding sensitive association rules with limited side effects," *IEEE Trans. Knowledge and Data Engineering*, vol. 19, issue 1, pp. 29–42, Jan. 2007.

[18] Y. Saygin, V. S. Verykios, and A. K. Elmagarmid, "Privacy preserving association rule mining," in *Proc. 12th Int. Workshop Research Issues in Data Mining Engineering, E- Commerce and E-Business Systems*, USA, Feb. 2002, pp. 151 – 158.

[19] C. Weng, S. Chen, and H. C. Lo, "A novel algorithm for completely hiding sensitive association rules," *IEEE Int. Conf. Intelligent Systems Design and Applications*, USA, vol. 3, pp. 202-208, Nov. 2008.

[20] S. R. M. Oliveira and O. R. Zaiane, "Privacy preserving frequent itemset mining," in *Proc. IEEE Workshop on Privacy, Security and Data Mining*, Dec. 2002, pp. 43 – 54.

[21] S. Oliveira and O. Zaiane, "Protecting sensitive knowledge by data sanitization," in *Proc. the 3rd IEEE Int. Conf. Data Mining*, USA, Nov. 2003, pp. 613-616.

[22] G. Lee, C. Y. Chang, and A. L. P. Chen, "Hiding sensitive patterns in association rules mining," in *Proc. the 28th IEEE Annual. Int. Computer Software and Applications*, USA, Sept. 2004, pp. 424-429.

[23] M. N. Dehkordi, K. Badie, and A. K. Zadeh "A novel method for privacy preserving in association rule mining based on genetic algorithms," *Journal of Software*, vol. 4, no. 6, pp. 555-562, Aug. 2009.

[24] S. Narmadha and S. Vijayarani, "Protecting sensitive association rules in privacy preserving data mining using genetic algorithms," *International Journal of Computer Applications*, vol. 33, no. 7, pp. 37-34, Nov. 2011.

[25] R. A. Shah and S. Asghar, "Privacy preserving in association rules using genetic algorithm," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 22, issue 2, pp. 434-450, March 2014.

[26] P. RajyaLakshmi, C. M. Rao, M. Dabbiru, and K. V. Kumar, "Sensitive itemset hiding in multi-level association rule mining," *International Journal of Computer Science & Information Technology*, vol. 2, no. 5, pp. 2124-2126, Sept-Oct. 2011.

Saad M. Darwish received his Ph.D. degree from the Alexandria University, Egypt. His research work concentrates on the field of image processing, optimization techniques, security technologies, computer vision, pattern recognition and machine learning. Dr. Saad is the author of more than 40 articles in peer-reviewed international journals and conferences and severed as TPC of many international conferences. Since Feb. 2012, he has been an associate professor in the Department of Information Technology, Institute of Graduate Studies and Research, Egypt.

Magda M. Madbouly received her Ph.D. degree from the Alexandria University, Egypt. Her research and professional interests include Artificial intelligence, cloud computing, neural networks and machine learning. She is an assistant professor in the Department of Information Technology, Institute of Graduate Studies and Research, Egypt.

Mohamed A. El-Hakeem received the B.Sc. degree in accounting from the Faculty of Commerce, University of Alexandria, Egypt in 2008. He has been a teaching assistant in the Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Egypt. His research and professional interests include database processing, data mining and security technologies.