Integrated Association Rules Complete Hiding Algorithms

Mohamed Refaat ABDELLAH[1], *Hesham Aboelsoud MOHAMED*[1], *Khaled Shafee BADRAN*[1], *Mohamed Badr SENOUSY*[2]

[1]Department of Computer Engineering, Military Technical College, El-Qobba Bridge, Al Waili, Cairo, Egypt
[2]Department of Computer and Information Systems, Sadat Academy, Street 151, Maadi Al Khabiri Al Wasti, Al Maadi, 12411 Cairo, Egypt

m_refaat_m@hotmail.com, h_aboelsoud@mtc.edu.eg, khaledbadran@mtc.edu.eg, badr_senousy_arcoit@yahoo.com

DOI: 10.15598/aeee.v15i2.2164

Abstract. *This paper presents database security approach for complete hiding of sensitive association rules by using six novel algorithms. These algorithms utilize three new weights to reduce the needed database modifications and support complete hiding, as well as they reduce the knowledge distortion and the data distortions. Complete weighted hiding algorithms enhance the hiding failure by 100 %; these algorithms have the advantage of performing only a single scan for the database to gather the required information to form the hiding process. These proposed algorithms are built within the database structure which enables the sanitized database to be generated on run time as needed.*

Keywords

Association rule, complete hiding approaches, database security, privacy preserving data mining.

1. Introduction

Hiding the sensitive rules, not the sensitive data, is the main objective of association rule hiding [1], which is done by sanitizing the data so that the association rule mining algorithms can extract all the non-sensitive rules and un-extract the sensitive rules. Sanitization is done by making some changes in the original data set. Complete hiding means the capability to hide all the sensitive association rules (zero hiding failure). This paper is organized as follows; association rule hiding process and related work are discussed in Sec. 2. and Sec. 3. , respectively. The proposed solution and experiments results are explained in Sec. 4. and Sec. 5. , respectively. Finally, conclusions explanation is included in Sec. 6.

2. Association Rule Hiding Process

2.1. Problem Description

The general definition of the problem is that we have a transnational dataset (database) D that contains sensitive information which needs to be protected from inference. Applying association rule mining algorithm to this dataset generates a set of association rules R with algorithm parameters Minimum Confidence Threshold (MCT) and Minimum Support Threshold (MST).

R is divided into two subsets: a set of the sensitive rules R_{sen} that needs to be protected, and a set of the non-sensitive rules $R_{non-sen}$. The problem solution is to generate the sanitized database D', which when encountered to rule mining techniques generates a new set of association rules R'. This new set is divided into a set of non-sensitive association rules $R'_{non-sen}$ and a set of sensitive rules that could not be hidden $R_{non-Hide}$, and a set of lost non-sensitive rules that were not meant to hide Fig. 1 demonstrates the association rule hiding rule sets [5].

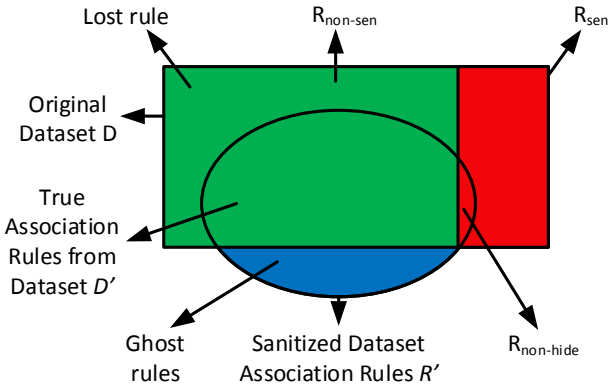

Fig. 1: Association rule hiding process.

2.2. Problem Formulation

The following notations are used to clarify the problem formulation as follows:

- $I = i_1, i_2, \ldots, m$: a set of finite m literals. Each member of I is called an item,
- X is the item set, where $X \subseteq I$,
- t: transaction is the set of items, where $t = \{i_k \mid i_k \in I, k \leq m\}$,
- The relation between database and transactions is given by $D = \{t_1, t_2, \ldots, t_n \mid n \in N\}$,
- The item set is supported by a transaction if $X \subset I$ and $t \in D$ if $X \subseteq t$.

$sup(X)$: support of X, which is the frequency of an itemset X in the database, and it is defined as:

$$sup(X) = |X(t)|, \tag{1}$$

where $X(t) = \{t \in D \mid t \,\text{contains}\, X\}$. If $sup(X) \geq MST$ then the itemset X is described as a frequent itemset. An association rule is represented as $I_L \rightarrow I_R$, where $I_L \cap I_R = \Phi$ and I_L, $I_R \subset I$, where I_R is the RHS (Right Hand Side) itemset and I_L is the LHS (Left Hand Side) itemset. The support of a rule $I_L \rightarrow I_R$ is the support of itemset $I_L \cup I_R$, as the Eq. (2) [13],

$$sup(I_L \rightarrow I_R) = sup(I_L \cup I_R). \tag{2}$$

The rule $I_L \rightarrow I_R$ confidence is defined as [13].

$$conf(I_L \rightarrow I_R) = \frac{sup(I_L \cup I_R)}{sup(I_L)}. \tag{3}$$

The association rule $I_L \rightarrow I_R$ is called strong association rules If $sup(I_L \rightarrow I_R) \geq MST$ and $conf(I_L \rightarrow I_R) \geq MCT$ Apriori property [13]: If $I_L, I_R \subseteq I$, and $I_L \subseteq I_R$, then $sup(I_L) \geq sup(I_R)$. This means that if an itemset I_L is frequent, then all itemsets that are subsets of I_L are frequent. The main hiding approaches shown in the next Fig. 2 can be based on the following description:

- Decreasing the confidence as the ISL (Increase Support of LHS) and DSR (Decrease Support of RHS).
- Decreasing the support as DSL (Decrease Support of LHS).

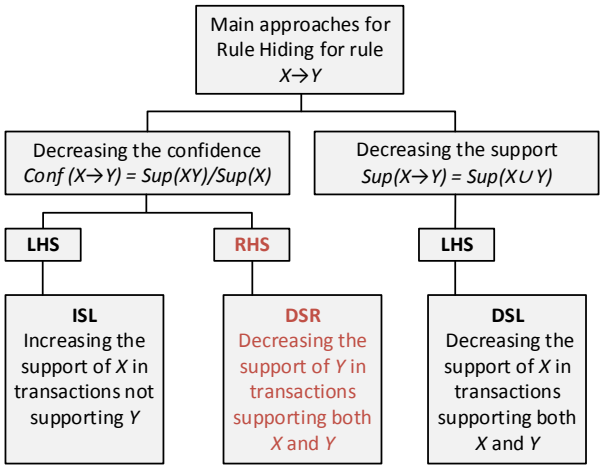

Fig. 2: The main hiding approaches.

2.3. Association Rule Hiding Measures

Association rule hiding algorithm performance is measured by commonly used methods in order to evaluate the proposed weighted algorithms.

- **Hiding Failure (HF)** HF measures the sensitive rules that are not hidden and can be mined from sanitized dataset. The hiding failure measurement is defined as the percentage of the sensitive data that remains discoverable in the sanitized dataset to the total number of sensitive rules to be hidden in the original dataset as shown in next equation [5]:

$$HF = \frac{S_R(D')}{S_R(D)}, \tag{4}$$

where D is the original data set, D' is the sanitized data set, and S_R is the number of sensitive association rules.

- **Misses Cost (MC)** MC measures the amount of non-sensitive association rules (lost rules) that are hidden by accident after sanitization, It is calculated by counting the non-sensitive data hidden after the sanitization process $(S'_R(D) - S'_R D')$ and dividing it by the all non-sensitive rules in the original dataset $D(S'_R(D))$, using the following formula [5]:

$$HC = \frac{S'_R(D) - S'_R(D')}{S'_R(D)}. \tag{5}$$

- **Artificial Patterns (AF)** AF measures the artificial association rules (ghost rules) that cannot be extracted from the original dataset but it can be extracted from sanitized dataset [11], which is created during the sanitization process due to the addition of noise in the data, and is calculated by:

$$AF = \frac{|R| - |R \cap R'|}{|R'|}, \tag{6}$$

where R is the set of discovered association rules in the original database D, R' is the set of association rules in the sanitized database D', and $|X|$ denotes the cardinality of X. $|X|$ is described as the cardinality of X.

- **Knowledge Distortion (KD)** KD is the total knowledge distortion. It is calculated as the cumulative sum of the amount of missing non-sensitive rules (Misses Cost MC) and the amount of ghost rules (Artificial Patterns AF) [11] as show in Eq. (7).

$$KD = MC + AF. \tag{7}$$

- **Data Distortion (DD)** DD measures the difference between sanitized database and original database.

$$DD = \frac{|T_{vi}|}{|T_n|}, \tag{8}$$

where $|T_{vi}|$ is the number of victim transactions that are modified in dataset D_in order to hide the sensitive rules. T_N is the total number of dataset transactions [11].

3. Related Work

In 2005, S. Wang et al. [2], proposed the DSR (Decrease Support of RHS) algorithm and the ISL (Increase Support of LHS) algorithm. DSR decreases the sensitive rule support and confidence below MST, MCT respectively to hide it. ISL works by rising support of sensitive rule LHS to hide it; confidence will be reduced under the MCT. DSR result shows no hiding failure; while ISL may fail when there are no appropriate transactions to add. In 2010, Modi et al. [3], created a new algorithm DSRRC (Decrease Support of RHS Items of Rule Cluster) to reduce hiding side effects by grouping the sensitive rules by similarity of RHS before the start the hiding process. This algorithm has two side effects:

- it increases the execution time due to the needed ordering of the database after each changes,
- it does not maintain data quality.

In 2011, Jain et al. [4], proposed a new algorithm that hides the rule by reducing and increasing the support of the RHS and LHS item of the rule at the same time. The advantage of this algorithm is its utilization of lower processing power than the previous work as a result of minimization of the data updates needed to hide a set of rules. In 2012, Shah et al. [6], proposed RRLR (Remove and Reinsert LHS of Rule) and ADSRRC (Advanced Decrease Support of RHS items of Rule Cluster) to enhance the performance of DSRRC. ADSRRC and DSSRC group sensitive rules by using the same RHS. ADSRRC is faster than DSSRC because it started with sorting the transactions according to the sensitivity in descending order. RRLR can hide association rules with multiple RHS. In 2012, Jain et al. [7] introduced a new method called Representative Rule (RR), where sensitive rules can be hidden without major changes in database. It is based on altering the position of items so the frequent itemsets support is still the same. The side effect of RR is the confidence computation for the non- strong rules that has confidence lower than MCT. In 2013, Domadiya et al. [8], proposed Modification Decrease Support of RHS items of Rule Clusters (MDSRRC). It can hide rules with multiple items in LHS and RHS. It begins with deleting items with highest values of sensitive rule items based on RHS. This decreases the database modification. MDSRRC has more benefits than DSRRC, including less side effects and improved data quality. In 2013, Dhutraj et al. [9], introduced a new algorithm using both DSR and ISL methods. This algorithm has two disadvantages:

- it cannot hide association rule with multiple items in RHS and LHS,
- high memory usage.

In 2014, Cheng et al. [10] proposed a hybrid algorithm that uses data distortion algorithm with a genetic algorithm named Evolutionary Multiobjective Optimization (EMO). The selection of deleting items needs more effort. It can effectively hide sensitive rules while generating fewer side effects. But it suffers from high count lost rules. In 2015, Cheng et al. [11] proposed improved hybrid algorithm to EMO by changing the hiding method from deleting items to Adding Items. It is called EMO-AddItem algorithm (HypE). It can do the hiding task with less knowledge distortions for most test cases.

4. Proposed Solution

In this work, a program is implemented for Apriori algorithm which is the most popular algorithm to find all the frequent sets and learn association rules. This

program is used to generate the association rules from the dataset and verify the results by using Waikato Environment for Knowledge Analysis (WEKA) Apriori associations tool [14]. In addition, six weighted hiding algorithms were designed as follows:

- W_ISL: Weighted Increase Support of LHS,
- W_DSL: Weighted Decrease Support of LHS,
- W_DSR: Weighted Decrease Support of RHS,
- W_C_DSL_DSR_C: Weighted Complete Hiding by integrating W_DSL and W_DSR using minimum changes method,
- W_C_ISL_DSR_C: Weighted Complete Hiding by integrating W_ISL and W_DSR using minimum changes method,
- W_C_ISL_DSR_S: Weighted Complete Hiding by integrating W_ISL and W_DSR using minimum Sensitive Rule Weight SRW method.

Algorithms W_ISL, W_DSL and W_DSR support the complete hiding with certain conditions depending on the database, the sensitive rules and the algorithm itself. These algorithms, W_C_DSL_DSR_C, W_C_ISL_DSR_S, and W_C_ISL_DSR_C, respectively, support the complete hiding for all sensitive rules.

4.1. Results Validations

Validation performed by comparing results of the proposed algorithms and the Algo1.a (Based on Increasing the support of the left hand side) [15], WSDA (Weight based Sorting Distortion Algorithm) [16], SIF-IDF (sensitive items frequency-inverse database frequency) [17], and the EMO-AddItem (Multi-Objective Optimization (EMO) based on many objective optimizations that using Adding Items) algorithm proposed on the work of Cheng et al. 2015 [11]. All results are calculated for the same MST = 5 % and MCT = 50 %, by using the same dataset, same sensitive rules, and same Apriori settings applied in Cheng et al. 2015.

4.2. Utilizing Victim Transaction Weights in the Invented Six Weighted Algorithms:

- **Transaction Frequent Rule Weight TFRW:** Each victim transaction Vi is assigned a transaction weight TFRW(Vi) calculated as the count of the frequent and non-sensitive rules that is fully supported by transaction Vi.
- **Non-sensitive Rules Weight NSRW:** Each victim transaction Vi is assigned a non-sensitive rules weight NSRW(Vi) which is calculated as the count of the frequent and non-sensitive rules supported by transaction Vi and can be hidden while applying sensitive rule hiding changes.
- **Sensitive Rules Weight SRW:** Each victim transaction Vi is assigned to a sensitive rules weight SRW(Vi) which is calculated as the count of sensitive rules that can be hidden by using transaction Vi with a hiding method.

4.3. Reuse victim transactions RVT:

RVT is a new method that collects all possible victim transactions for all sensitive rules, and then allows the hiding algorithm to select the victim transactions from this collection. Then it applies the selection method to support reusing of the victim transaction to hide more than one sensitive rule. It is used with SRW(Vi) weight to reduce the transactions data distortion and total database modifications.

4.4. Basic Notation and Definitions

Let $sup(R)$ be the initial support of the rule R. Let $conf(R)$ be the initial confidence of the rule R. Let T_L be the transactions that support I_L. Let $|T_L|$ be the count of the transactions that support I_L. Let SSM be the Support safety Margin threshold and it is used with DSL method. Let CSM be the Confidence Safety Margin threshold and it is used with DSR (CSM_DSR) and ISL (CSM_ISL) methods. Let $Csup(R)$ be the minimum changes needed to hide a rule by changing the rule support and represents the minimum transactions count needed to decrease the $sup(R)$ in order to hide rule R. Let $Cconf(R)$ to be the minimum changes needed to hide a rule by changing the rule confidence and I represents the minimum transactions count needed to decrease the $conf(R)$ in order to hide rule R. Transactions which are updated to hide the rule R are called rule victims $victim(R)$.

1) For W_DSR method:

$$conf(R_i) = \frac{sub(R_i)}{|T_L|}, \quad \frac{sub(R_i) - Cconf(R_i)}{|T_L|} \leq \leq (MCT - CSM\_DSR), \tag{9}$$

$$Cconf(R_i) = sub(R_i) - (MCT - CSM\_DSR) - |T_L|. \tag{10}$$

This W_DSR confidence hiding minimum changes is applied on transactions that fully support I_L and I_R $|T_{LR}|$. The condition for W_DSR complete hiding is defined as $|T_{LR}| \geq Cconf(R_i)$, where $|T_{LR}|$ is the count of transactions that fully support I_L and I_R.

2) For W_ISL method:

$$conf(R_i) = \frac{sub(R_i)}{|T_L|}, \quad \frac{sub(R_i) - Cconf(R_i)}{|T_L| + Cconf(R_i)} \leq \leq (MCT - CSM\_ISL), \tag{11}$$

$$Cconf(R_i) = \frac{sub(R_i)}{(MCT - CSM\_ISL)} - |T_L|. \tag{12}$$

This W_ISL confidence hiding minimum changes is applied on the transactions that partially support I_L but do not support I_R (T_{LpRn}).The condition for W_ISL complete hiding is$|T_{LpRn}| \geq Cconf(R_i)$, where $|T_{LpRn}|$ is the count of transactions that partially support I_L but do not support I_R.

3) For W_DSL method:

$$Cconf(R_i) = sub(R_i) + -((MST - SSM) \cdot |T_N|). \tag{13}$$

This W_DSL confidence hiding minimum changes is applied on transactions that fully support I_L and I_R (T_{LR}). The condition for W_DSL complete hiding is defined as $|T_{LR}| \geq Cconf(R_i)$, where $|T_{LR}|$ is the count of transactions that fully support I_L and I_R.

4) For W_C_DSR method:

The complete hiding condition is $|T_{LR}| < Cconf(R_i)$ so we need to increase the transactions that fully support I_L and I_R T_{LR} by making $(Cconf(Ri) - |T_{LR}|)$ changes in transaction not supporting I_L and supporting I_R.

5) For W_C_ISL method:

The complete hiding condition is $|T_{LpRn}| < Cconf(R_i)$ so we need to increase transactions that partially support I_L but do not support I_R (T_{LpRn}) by making $(Cconf(R_i) - |T_{LpRn}|)$ changes in transaction not supporting I_L and supporting I_R.

6) For W_C_DSL method:

The complete hiding condition is $|T_{LR}| < Cconf(R_i)$so we need to increase the transactions that fully support I_L and I_R (T_{LR}) by making $(Cconf(R_i) - |T_{LR}|)$ changes in transaction supporting I_L and not supporting I_R.

4.5. Victim Transaction Weights:

1) Transaction Frequent Rule Weight TFRW

"The lower the best" Each victim transaction Vi assigns a transaction weight $W(Vi)$ which is calculated as the count of frequent and non-sensitive rules that are fully supported by transaction Vi, where $|RF(Vi)|$ is the count of frequent and non-sensitive rules that are fully supported by victim transaction Vi.

$$TFRW(Vi) = |RF(Vi)|. \tag{14}$$

2) Non-sensitive Rules Weight NSRW

"The lower the best"Each victim transaction Vi assigns a non-sensitive rules weight NSRW (Vi) and it is calculated as the count of frequent and non-sensitive rules supported by transaction (Vi) and can be hidden while applying sensitive rule hiding changes.

$$NSRW(Vi) = = \text{Non\_sens\_Rules}(Vi, \text{hiding method}). \tag{15}$$

For W_DSR,the algorithm selects all frequent and non-sensitive rules that are fully supported by transaction (Vi) and RHS of the non-sensitive rule same as I_R for sensitive rule R_i.

For W_DSL algorithm selects all the frequent and non-sensitive rules that are fully supported by transaction Vi and LHS of the non-sensitive rule same as I_L for sensitive rule R_i. For W_ISL algorithm select all the frequent and non-sensitive rules that are fully supported by transaction Vi and LHS of the non-sensitive rule same as I_L new item values for sensitive rule R_i.

3) Sensitive Rules Weight SRW

"The higher the best" Each victim transaction $(Vi)$is assigned sensitive rules weight SRW(Vi) which is calculated as the count of sensitive rules that can be hidden by using transaction Vi with a hiding method.

$$SRW(Vi) = \text{Sens\_Rules}(Vi, \text{hidingmethod}). \tag{16}$$

where Sens_Rules function is calculated with respect to given hiding method.

4.6. Proposed Weighted Hiding Association Rule Algorithms

In this section, all proposed algorithms are explained. The inputs and outputs of algorithms are summarized as follow: Algorithm Inputs are defined as:

- a finite transaction database D,
- MST = 5 % and MCT = 50 %. SSM for DSL method = 0.0001, CSM for ISL method = 0.0008 and CSM for DSR method = 0.0009,
- the set R_{sen} of sensitive rules (10 Rules). Algorithm Output: A sanitized database D'.

1) W_ISL: Weighted Increase Support of LHS

This method is based on increasing the support of sensitive rule LHS by updating the selected transactions that partially support rule LHS and do not support rule RHS. The complete hiding is achieved if the number of available transactions is higher than or equal to the hiding minimum $changesCconf(R_i)$.

2) W_DSL: Weighted Decrease Support of LHS

This method is based on decreasing the support of sensitive rule LHS by updating the selected transactions that fully support rule LHS and RHS. The complete hiding is achieved if the number of available transactions is higher or equal to the hiding min. $changesCsup(R_i)$.

3) W_DSR: Weighted Decrease Support of RHS

This method is based on decreasing the support of sensitive rule RHS by updating selected transactions that fully support rule LHS and RHS. The complete hiding is achieved if the number of available transactions is higher than or equal to the hiding min. $changesCconf(R_i)$.

Main Steps of Weighted hiding algorithm: Main steps and difference when applied the RVT method are shown in Fig. 2 and Fig. 3 are explained as follows:

- calculate the hiding minimum changes required for hiding by this method,
- calculate the TFRW(Vi), SRW(Vi), and NSRW(Vi) for all victims transactions of this method,

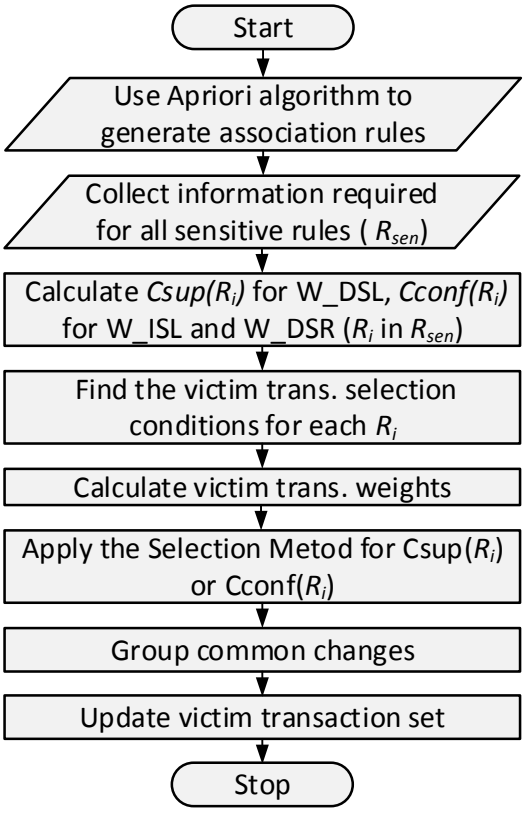

Fig. 3: Weighted hiding algorithms.

- for each sensitive rule R_i in R_{sen} selects the number of transactions equal to the hiding minimum $changes(Cconf(Ri)$ for W_ISL, W_DSR or $Csup(Ri)$ for W_DSL) order by NSRW(Vi) ascending, SRW(Vi) descending, and TFRW(Vi) ascending,
- get the final transaction changes set by group common transactions for all sensitive rules where common transaction means transaction that used to hide more than one sensitive rule,
- update the database by the final transaction changes set to get the sanitized DB D'.

Notes:

- if the available transaction in step 3 is less than the hiding changes minimum, then hiding failure for this sensitive rule occurs,
- we test different order by methods for step 3 like TFRW ascending, SRW(Vi) descending or SRW(Vi) descending and TFRW ascending,
- SRW(Vi) and grouping transactions in Step 4 are used to reduce the transactions data distortion and total database modifications,
- NSRW(Vi) or TFRW(Vi) or both are used to reduce the knowledge distortion.

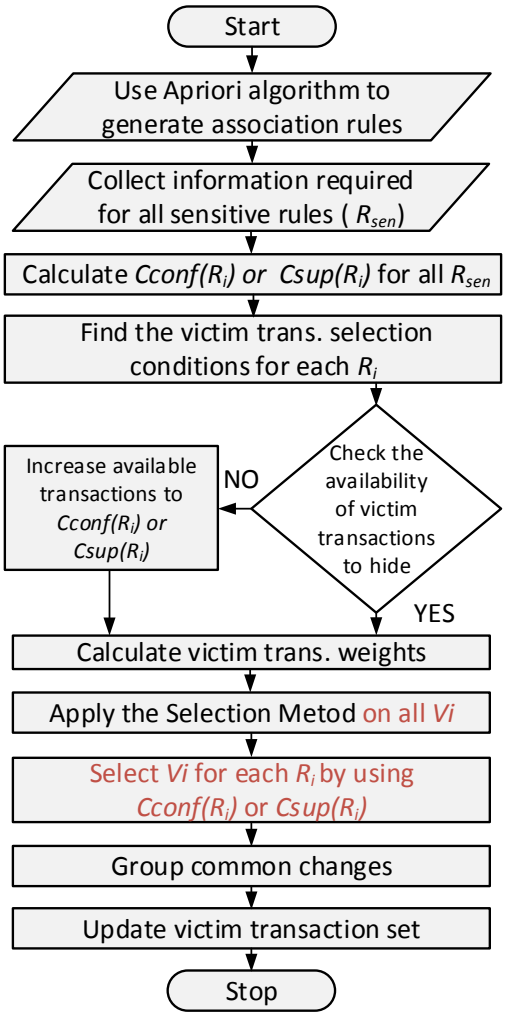

Fig. 4: Weighted hiding algorithms with RVT.

4) W_C_DSL_DSR_C

By using this method, we are able to calculate the hiding minimum changes for W_DSL and W_DSR, and check the availability of enough transaction for complete hiding, then select hiding method for each sensitive rule such that it supports complete hiding and achieves the minimum changes when comparing the two methods.

Steps of the algorithm:

- calculates $Cconf(R_i)$ for W_DSR and $Csup(R_i)$ for W_DSL,
- selects hiding method for each sensitive rule R_i based on support of complete hiding and minimum changes,
- calculates the TFRW (Vi), SRW(Vi) and NSRW(Vi) for all victims transactions of R_i rules hiding by W_DSL method,
- calculate the TFRW (Vi), SRW(Vi), and NSRW(Vi) for all victims transactions of R_i rules hiding by W_DSR method,
- for each sensitive rule R_i in R_{sen} selects the number of transactions equal to the hiding minimum changes required for its hiding method. This selection is done based on ordering all suitable transactions by NSRW(Vi) ascending, SRW(Vi) descending and TFRW ascending,
- get the final transaction change set by group common transactions for all sensitive sets.

Update the database by the final transaction change set to get the sanitized database D'.

5) W_C_ISL_DSR_C

By using this method, we are able to calculate the hiding minimum changes for W_ISL and for W_DSR, and check the availability of enough transaction for complete hiding, then select hiding method for each sensitive rule such that it supports complete hiding and achieves the minimum changes when comparing the two methods.

Steps of algorithm:

- calculate $Cconf(Ri)$ for both methods W_ISL and for W_DSR,
- select hiding method for each sensitive rule Ri based on support of complete hiding and minimum changes,
- calculate the TFRW (Vi), SRW(Vi), and NSRW(Vi) for all victims transactions of R_i rules hiding by W_DSR method,
- for each sensitive rule R_i in R_{sen} select the number of transactions equal to the hiding minimum changes required for its hiding method. This selection is done based on ordering all suitable transactions by NSRW(Vi) ascending, SRW(Vi) descending and TFRW ascending,
- get the final transaction change set by group common transactions for all sensitive sets,
- update the database by the final transaction change set to get the sanitized database D'.

6) DB_C_ISL_DSR_S

By using this method, we are able to calculate the hiding minimum changes for W_ISL for W_DSR and check the availability of enough transaction for complete hiding. The selection of the hiding method for

Tab. 1: Characteristics of Mushroom dataset and parameter setting.

Transaction No.	Items Avg.	Transaction Length	MST	MCT	Frequent Itemsets No.	Strong Rules No.
8124	119	23	5 %	50 %	1329	1065

Tab. 2: Weighted hiding algorithms results.

Hiding Measure	EMO-AddItem	Algo1.a	WSDA	SIF-IDF	Proposed Algorithms		
					W_ISL	W_DSL	W_DSR
HF %	20	20	0	0	20	0	0
KD %	2.449	5.087	2.935	8.431	1.706	10.616	4.076
DD %	49.489	36.234	36.148	26.105	33.112	31.45	24.975

Tab. 3: Integrated weighted complete hiding algorithms results.

Hiding Measure	WSDA	SIF-IDF	Proposed Algorithms				
			W_ISL	W_C_ISL_DSR_C	W_C_ISL_DSR_S	W_DSL	W_C_DSL_DSR_C
HF %	0	0	20	0	0	0	0
KD %	2.935	8.431	1.706	1.517	1.137	10.616	9.953
DD %	36.148	26.105	33.112	31.425	29.554	31.45	27.536

each sensitive rule is based on its support of the complete hiding and have the minimum SRW from the two methods.

Steps of algorithm:

- calculate $Cconf(R_i)$ for both methods W_ISL and for W_DSR,
- select hiding method for each sensitive rule Ri based on support of complete hiding and minimum SRW(R_i),
- calculate the TFRW (Vi), SRW(Vi), and NSRW(Vi) for all victims transactions of R_i rules hiding by W_ISL method,
- calculate the TFRW (Vi), SRW(Vi), and NSRW(Vi) for all victims transactions of R_i rules hiding by W_DSR method,
- for each sensitive rule R_i in R_{sen} select the number of transactions equal to the hiding minimum changes required for its hiding method. This selection is done based on ordering all suitable transactions by NSRW(Vi) ascending, SRW(Vi) descending and TFRW ascending,
- get the final transaction change set by group common transactions for all sensitive sets,
- update the database by the final transaction change set to get the sanitized database D'.

5. Experiments Results and Analysis

5.1. Experimental Setup

The proposed approaches use the oracle database 11g, Procedural Language/Structure Query Language PL/SQL 11.0.2.0, and run on an Intel i5 CPU 660 with four processors with 3.33 GHz speed and main memory with 4 GB. We did extensive experiments on real dataset. the experimental results are based on the following measures:

- hiding failure: The amount of sensitive rules that fail to be hidden (The lower the better),
- knowledge distortion: it is the sum of the two measures of lost non-sensitive rules and ghost rules (The lower the better),
- data distortion (Data loss): it is the amount of transactions changes needed to obtain the sanitized database (The lower the better).

5.2. Used Dataset

We examined the proposed algorithms prepared by Roberto Bayardo using the Mushroom dataset which was to represent the real database. The Characteristics of Mushroom dataset and parameter settings [11] are shown in Tab. 1.

5.3. Weighted Hiding Algorithms Results: Tab. 2

Weighted hiding algorithms results analysis: Since the HF measurement is of higher priority in the evaluation of any hiding algorithm, this work compares the proposed weighted algorithm with the other evaluated algorithms on the criteria of HF percentage value. We compare EMO-AddItem and Algola with the proposed algorithm W_ISL since they have the HF = 20 %, similarly WSDA and SIF-IDF algorithm compared with W_DSR for HF = 0 %. The W_algorithm has high KD measurement value with respect to all the proposed algorithms. For HF = 20 %, W_ISL evaluated by EMO-AddItem shows that KD is enhanced by 30 % and DD was enhanced by 30 %, while for The W_ISL algorithm when compared to Algo1.a, KD was enhanced by 66 % and DD was enhanced by 9 %. In case of the complete hiding algorithms W_DSR compared to SIF-IDF, KD had improved by 52 % and DD had improved by 4 %, respectively. When W_DSR is compared to WSDA, the KD value was increased by 38 % and DD had enhanced by 31 %.

5.4. Integrated Weighted Complete Hiding Algorithms Results: Tab. 3

Integrated weighted complete hiding algorithms results analysis: The W_C_ISL_DSR_C and W_C_ISL_DSR_S successfully the complete hiding for the W_ISL algorithm and enhance both KD and DD. The W_C_ISL_DSR_S compared to W_ISL was HF enhanced by 100 %; KD enhanced by 33 % and DD enhanced by 11 %. The W_C_ISL_DSR_S compared to WSDA has KD enhanced by 61 % and DD enhanced by 18 %; similarly W_C_ISL_DSR_S compared to SIF-IDF has KD improved by 87 % and DD increased by 13 %. The W_C_DSL_DSR_C successfully enhanced both KD and DD by 6 % and 12 %, respectively.

6. Conclusions

The novel improved algorithms show better results compared to EMO-AddItem, Algo1.a, WSDA, and SIF-IDF. The results of weighted algorithms W_ISL, W_DSL, and W_DSR show that only W_ISL does not support the complete hiding because the algorithm hiding results.

depend on the hiding methods and its implementations in the hiding algorithm, sensitive rules, and dataset. So we proposed the integrated algorithms to achieve the complete hiding. The integrated algorithms W_C_ISL_DSR_C, and W_C_ISL_DSR_S achieve the complete hiding for the W_ISL algorithm and enhance both KD and DD measures. The W_C_DSL_DSR_C algorithm enhanced the KD and DD measures for W_DSL.

The use of grouping of common victim transactions and SRW, TFRW selection method achieved lower data distortion with W_ISL, W_DSR algorithms. The change of the selection methods enhances the knowledge distortion KD or the data distortion DD measures and can be chosen according to the problem requirements. Those algorithms only need a single scan of the database to hide the sensitive rules so the victim transaction weights applied on all victim transactions which help to be more effective. Integration of this algorithm to the database structure adds new capability to generate the sanitized database in the run time when required.

References

[1] FARKAS, C. and S. JAJODIA. The inference problem: A survey. *ACM SIGKDD Explorations Newsletter*. 2002, vol. 4, iss. 2, pp. 6–11. ISSN 1931-0145. DOI: 10.1145/772862.772864.

[2] WANG, S.-L. and A. JAFARI. Hiding sensitive predictive association rules. In: *International Conference on Systems, Man and Cybernetics*. Waikoloa: IEEE, 2005, pp. 164–169. ISBN 0-7803-9298-1. DOI: 10.1109/ICSMC.2005.1571139.

[3] MODI, C. N., U. P. RAO and D. R. PATEL. Maintaining Privacy and Data Quality in Privacy Preserving Association Rule Mining. In: *International Conference on Computing Communication and Networking Technologies (ICCCNT)*. Karur: IEEE, 2010, pp. 1–6. ISBN 978-1-4244-6592-7. DOI: 10.1109/ICCCNT.2010.5592589.

[4] JAIN, Y. K., V. K. YADAV and G. S.PANDAY. An Efficient Association Rule Hiding Algorithm for Privacy Preserving Data Mining. *International Journal on Computer Science and Engineering*. 2011, vol. 3, iss. 7, pp. 2792–2798. ISSN 0975-3397. Available at: http://www.enggjournals.com/ijcse/doc/IJCSE11-03-07-054.pdf.

[5] ABDELLAH, M. R., H. M. ABOELSEOUD, K. S. BADRA and M. B. SENOUSY. Privacy Preserving Association Rule Hiding Techniques: Current Research Challenges. *International Journal of Computer Applications*. 2016, vol. 136, no. 6, pp. 11–17. ISSN 0975-8887. DOI: 10.5120/ijca2016908446.

[6] SHAH, K., A. THAKKAR and A. GANATRA. Association Rule Hiding by Heuristic Approach to Reduce Side Effects and Hide Multiple R. H. S. Items. *International Journal of Computer Applications*. 2012, vol. 45, no. 1, pp. 1–7. ISSN 0975-8887. DOI: 10.5120/6741-7813.

[7] JAIN, D., P. KHATRI, R. SONI and B. K. CHAURASIA. Hiding Sensitive Association Rules without Altering the Support of Sensitive Item (s). In: *Advances in Computer Science and Information Technology (CCSIT)*. Berlin: Springer, 2012, pp. 500–509. ISBN 978-3-642-27299-8. DOI: 10.1007/978-3-642-27299-8_52.

[8] RAO, U. P. and N. DOMADIYA. Hiding sensitive association rules to maintain privacy and data quality in database. In: *3rd IEEE International Advance Computing Conference (IACC)*. Ghaziabad: IEEE, 2013, pp. 404–407. ISBN 978-1-4673-4529-3. DOI: 10.1109/CSNT.2014.86.

[9] DHUTRAJ, N., S. SASAN and V. KSHIRSAGAR. Hiding Sensitive Association Rule for Privacy Preservation. *IEEE Transactions on Knowledge and Data Engineering*. 2013, vol. 25, no. 1, pp. 1–3. ISSN 1041-4347.

[10] CHENG, P., J. S. PAN and C. W. L. HARBIN. Hiding Sensitive Association Rules without Altering the Support of Sensitive Item (s). In: *IEEE Congress on Evolutionary Computation (CEC)*. Beijing: IEEE, 2014, pp. 1108–1115. ISBN 978-1-4799-1488-3. DOI: 10.1109/CEC.2014.6900539.

[11] CHENG, P., J. S. PAN and C. W. LIN. Use HypE to Hide Association Rules by Adding Items. *PLoS ONE*. 2015, vol. 10, iss. 6, pp. 1–19. ISSN 1932-6203. DOI: 10.1371/journal.pone.0127834.

[12] MOGTABA, S. and E. KAMBAL. Association Rule Hiding for Privacy Preserving Data Mining. In: *Industrial Conference on Data Mining (ICDM)*. New York: Springer International Publishing, 2016, pp. 320–333. ISBN 978-3-319-41561-1. DOI: 10.1007/978-3-319-41561-1_24.

[13] LE, H. Q. and S. ARCH-INT. A Conceptual Framework for Privacy Preserving of Association Rule Mining in E-Commerce. In: *7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. Singapore: IEEE, 2012, pp. 1999–2003. ISBN 978-1-4577-2119-9. DOI: 10.1109/ICIEA.2012.6361057.

[14] Waikato Environment for Knowledge Analysis (Weka). Machine Learning Group at the University of Waikato. In: *The University of Wakiato* [online]. 2017. Available at: `http://www.cs.waikato.ac.nz/~ml/weka/index.html`.

[15] VERYKIOS, V. S., A. K. ELMAGARMID, E. BERTINO, Y. SAYGIN and E. DASSENI. Association rule hiding View Document. *IEEE Transactions on Knowledge and Data Engineering*. 2004, vol. 16, iss. 4, pp. 434–447. ISSN 1041-4347. DOI: 10.1109/TKDE.2004.1269668.

[16] VERYKIOS, V. S., E. D. PONTIKAKIS, Y. THEODORIDIS and L. CHANG. Efficient algorithms for distortion and blocking techniques in association rule hiding. *Distributed and Parallel Databases*. 2007, vol. 22, iss. 1, pp. 85–104. ISSN 1573-7578. DOI: 10.1007/s10619-007-7013-0.

[17] HONG, T. P., C. W. LIN, K. T. YANG and S. L. WANG. Using TF-IDF to hide sensitive itemsets. *Applied Intelligence*. 2013, vol. 38, iss. 4, pp. 502–510. ISSN 1573-7497. DOI: 10.1007/s10489-012-0377-5.

About Authors

Mohamed Refaat ABDELLAH received a Bachelor degree in engineering from the Military Technical College (MTC), Cairo, Egypt, in 1994 and got his Master's degree in engineering fromComputer Engineering department, Cairo University, Egypt, in 2011. He is currently a Ph.D. student in Computer Engineering Department in MTC. His research interests are in data mining, database security and Data Hiding Techniques.

Hesham Aboelsoud MOHAMED receiveda Bachelor degree and Masters in computer engineering from the MTC, Cairo, Egypt, in 1993 and 2000, respectively. He also received a Ph.D. degree in Systems and Biomedical Engineering, from Military Technical Collage, in 2006. He is currently a faculty member in the Department of Computer Engineering, MTC. His research interests are in Digital Image Processing, Computer System Security, Database Security and Data Hiding Techniques.

Khaled Shafee BADRAN received a Bachelor degree incomputer engineering and Mastersdegree from the MTC, Cairo, Egypt, in 1995 and 2000, respectively. He also received the Ph.D. degree in Electrical and Computer engineering from Sheffield University, UK, in 2009. He is currently a faculty member with the Department of Computer Engineering, MTC. His research interests are in data mining, semantic web and database security.

Mohamed Badr SENOUSY received the Bachelor degree in engineering from the MTC, Cairo, Egypt, in 1973. He also received the Masters and Ph.D. degrees from George Washington University, Washington DC,

USA in 1982 and 1985, respectively. He is currently a faculty member in the Department of Computer and Information System, Sadat Academy Cairo, Egypt. His research interests are operating systems, software engineering and computer security.