International Journal of Electrical and Power Engineering 2 (3): 154-165, 2008
ISSN: 1990-7958
© Medwell Journals, 2008

A Panglossian Solitary-Skim Sanitisation for Privacy Preserving Data Archeology

[1]J. Gitanjali, [1]Shaik Nusrath Banu, [2]J. Indumathi and [2]G.V. Uma
[1]School of Computer Science, Vellore Institute of Technology, Vellore, India
[2]Department of Computer Science and Engineering, Anna University,
Chennai-600 025, Tamilnadu, India

Abstract: Knowledge is preeminence and the more conversant we are about information burglarizes; we are a lesser amount of prone to fall prey to the malevolence hacker sharks of information technology. The information technology is eternally emerging and we are all beyond uncertainty infinitesimal gravels in a extraterrestrial aquatic of information. Knowledge is pre-eminence, but as humble users of the most modern technologies we are pitted with possessions that may even make us paranoid concerning usage of a computer. The goal of privacy-preserving data mining is to liberate a dataset that researchers can study without being able to identify sensitive information about any individuals in the data (with high probability). One technique for privacy-preserving data mining is to replace the sensitive items by unknown values. For many situations it is safer if the sanitization process consign unknown values as a substitute of fake values. This obscures the susceptible rules, whilst defending the punter of the data commencing false rules. In this study, we remove some items which are sensitive from the transactional database at the same time retaining knowledge for sharing information. In this research, we have adapted the one-scan SWA christening it as Panglossian Solitary-Skim Sanitization algorithm and used it for Privacy Preserving Data Archaeology. We have utilised the notion of disclosure threshold for each lone pattern to curb and provide an enhanced agility allowing an administrator to place disparate weights for varied rules. Our research overcomes the privacy breach problem of existing blocking sanitizing approaches. We have investigated how probabilistic and information theoretic techniques can be applied to this problem. More complete analysis of the effectiveness of this Panglossian Solitary-Skim Sterilization for Privacy Preserving Data Archeology (P3SPPDA) based technique and formal study of the problem has been made. Our experiment reveals that our algorithm is competent, scalable and achieves noteworthy enhancement in excess of the other approaches offered in the literature. Our preliminary domino effect point toward deterministic algorithms for privacy preserved data and accuracy by controlling disclosure of sensitive data and knowledge.

Key words: Disclosure threshold, non-sensitive, solitary-skim algorithm, Panglossian, privacy preserving data archeology, sanitsation, sensitive, sliding window algorithm

INTRODUCTION

Recently, there has been an escalating accent on exploratory scrutiny of very large datasets to discover useful patterns and/or correlations among attributes which is called as data mining. Privacy preserving data mining is accomplishing valid data mining results devoid of the wisdom of the core data values has been receiving concentration in the research society and beyond.

Data Archeology (Also Known as Knowledge Discovery in Databases (KDD), knowledge extraction, data analysis, pattern analysis, data dredging, Information Harvesting, business intelligence, etc.) in databases is defined as the extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) information or patterns from data in large databases. Data Archeology has been proven useful to support both decision-making processes and to promote social goals. However, the partaking of data has furthermore raised a number of ethical issues as those of privacy, data security and intellectual property rights. These datasets more often than not enclose tantalizing personal information, which inexorably gets uncovered to diverse parties. As a result confidentiality issues are persistently under the glare of publicity and the public discontent may well pressurize the employment of Data Archeology and all its profits. It is thus of great significance to develop passable security techniques for defensive secrecy of individual values used for Data Archeology.

Corresponding Author: J. Indumathi, School of Computer Science, Anna University, Chennai, India

SANITISATION-AN OVERVIEW

One must redact information as of the raw data prior to providing it. The redaction may possibly get one of two forms: summary or sanitization.

A summary is a finished analysis of the data in which the pertinent information is used to calculate statistics for instance counts, means and so forth. Characteristically, a gross depiction of the data is well-known to provide milieu. The Indian Bureau of Statistics presents summaries of raw data to the public and does so in such a way that the raw data cannot be redefined from the synopsis. Consequently the summary conceals all aspects of the raw data that the government department desires to restrain. A summary of the communal data mentioned above would name the state the quantity of houses purchased by clients from that state.

Sanitization takes the contradictory approach. The raw data is offered for others to analyze, however the data is altered so that sensitive items are suppressed. Medical records given to researchers are treated in this style. Diagnostic information, symptoms and treatments are given, but information identifying the exact patient, such as name, address, phone number and so forth, are redacted. A sanitized set of the corporate data mentioned above might consist of a list of purchasers, their addresses and the number of houses each purchased, but the names and addresses would be replaced by meaningless strings.

The remuneration of sanitization are that the beneficiary of the data can analyze the raw data and obtain statistics, or take activities, based upon the data itself sooner than the provider's summary of the data. Based upon the components of the data that are sanitized the recipient needs to derive information which is a serious setback. In our medical example, the recipient may want to check the prevalence of a particular disease with respect to geography. As the provider has suppressed.

Static sanitization: When the total set of data to be sanitized is accessible at the time of sanitization, the sanitization functions can be derived completely prior to the sanitization of the data.

Dynamic sanitization: When the total set of data to be sanitized is not obtainable at the time of sanitization, the sanitization function may change as the data becomes available and is sanitized.

Basic ways to sanitize objects are:

Deletion: Here, the objects to be sanitized are minimally deleted.

Fixed transformation: All occurrences of the object are replaced by a fixed string.

Variable transformation: Occurrences of the object are transformed in different ways depending upon the context and structure of the object. For example, translating an IP address into one value for FTP connections and a different value for HTTP messages, is an example of variable replacement. Replacing objects with random data is another.

Typed transformation: This is a form of variable transformation, except that the replacing objects are related when the types of the object being replaced are the same. For example, replacing all file names with a value generated by one cryptographic hash function and all IP addresses with addresses selected from the 10. network, would be an example of this.

As shown in Fig. 1 we transform a database into a new one that conceals some premeditated patterns (restrictive association rules) while preserving the general patterns and trends from the original database. The procedure of transforming an original database into a sanitized one is called data sanitization The sanitization process acts on the data to remove or hide a group of restrictive association rules that contain sensitive knowledge.

From non-sensitive information or unclassified data, one is able to infer sensitive knowledge, including personal information, facts, or even patterns that are not supposed to be disclosed (Clifton and Marks, 1996). Data owners of two or more companies settle on to courteously demeanor association rule mining on their datasets for their communal benefit. However, some of these companies may not want to share some premeditated patterns hidden within their own data (also called restrictive association rules) with the other parties. They would like to alter their data in such a way that these restrictive associations rules cannot be discovered.

The modus operandi of transforming an original database into a sanitized one is called data sanitization The sanitization process acts on the data to confiscate or conceal a group of restricted association rules that restrain sensitive knowledge. On the one hand, this approach faintly modifies some data, but this is flawlessly

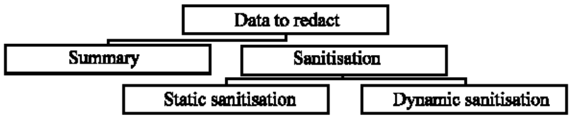


Fig. 1: Types of sanitization

tolerable in some real applications (Atallah *et al*., 1999; Dasseni *et al*., 2001; Saygin *et al*., 2001; Oliveira and Zaiane, 2002). On the other hand, an apposite poise flanked by a want for privacy and knowledge breakthrough must be cast-ironed.

The idea behind data sanitization was introduced by Atallah *et al*. (1999) considered the quandary of limiting disclosure of sensitive rules, aiming at selectively trouncing some frequent itemsets from large databases with as petite brunt on other non-sensitive frequent itemsets as possible. Specifically, the authors dealt with the crisis of modifying a known database so that the support of a given set of sensitive rules, mined from the database, decreases beneath the minimum support value. The authors focused on the conjectural approach and showed that the optimal sanitization is an NP-hard problem. Johnsten and Raghavan (1999) investigated secrecy issues of a broad class of association rules and proposed some algorithms to preserve privacy of such rules above a given privacy threshold. Inspite of ensuring privacy preservation these algorithms, are CPU-intensive due to multiple scans over a transactional database. In addition, such algorithms, in some way, modify true data values and relationships by turning some items from 0-1 in some transactions. In the equivalent route, Saygin *et al*. (2001) introduced a mode for selectively removing individual values from a database to prevent the discovery of a set of rules, while preserving the data for other applications. They proposed some algorithms to brume a given set of sensitive rules by replacing known values with unknowns, while minimizing the side effects on non-sensitive rules. These algorithms are CPU-intensive and require various scans depending on the number of association rules to be hidden.

Oliveira and Zaiane (2002) introduced a unified framework that combines techniques for efficiently muking restrictive patterns: a transaction retrieval engine relying on an inverted file and Boolean queries; and a set of algorithms to “sanitize” a database. In this framework, the sanitizing algorithms require two scans regardless of the database size and the number of restrictive patterns that must be protected. The first scan is required to build an index (inverted file) for speeding up the sanitization process, while the second scan is used to sanitize the original database.The introduced efficient one-scan algorithm, called Sliding Window Algorithm (SWA) (Olivera and Zaiane, 2003a, b) regardless of the database size and the number of restrictive association rules that must be protected the algorithm and needs only one pass over a transactional database. This represents a noteworthy enhancement over the previous algorithms presented in the literature (Dasseni *et al*., 2001; Berry and Linoff, 1991; Saygin *et al*., 2001; Oliveria and Zaiane, 2002) which require various scans depending on the number of association rules to be hidden. SWA is not a memory-based algorithm and therefore can deal with very large databases.

SWA somewhat alters the data and enables litheness for somebody to fine-tune it. SWA does not launch sham drops to the data; it has the tiniest misses price tag among the known sanitizing algorithms. Since there is no encryption implicated, the sanitization method is strong and no de-sanitization possible. There is no possible way to reproduce the original database from the sanitized one. As a result, the efficacy of the data, at the end of the privacy preserving process, is an imperative question, because in order for sensitive information to be hidden, the database is modified using SWA and the efficacy of the data falls because the less the database reflects the domain of interest. Therefore, an evaluation parameter for the data efficacy should be the amount of information that is lost after the application of privacy preserving process. Information loss in the context of association rule mining will be measured either in terms of the number of rules that were both remaining and lost in the database after sanitization, or even in terms on the reduction/increase in the support and confidence of all the rules.

Consequently, we have adapted the one-scan SWA christening it as Panglossian Solitary-Skim Sanitization algorithm and used it for Privacy Preserving Data Archaeology. We have utilized the notion of disclosure threshold for each lone pattern to curb and provide an enhanced agility allowing an administrator to place disparate weights for varied rules. Our experiment reveals that our algorithm is competent, scalable and achieves noteworthy enhancement in excess of the other approaches offered in the literature. We judge our proposed algorithm with the similar counterparts in the literature.

PROBLEM OF PANGLOSSIAN SOLITARY-SKIM SANITIZATION FOR PRIVACY PRESERVING DATA ARCHEOLOGY

We recommend a Panglossian Solitary-Skim Sanitization for Privacy Preserving Data Archeology algorithm based technique to disinfect a database into a new one that conceals some strategic patterns (restrictive association rules) while preserving the general patterns and trends. Our (P3SPPDA) based technique attempts to unearth poise amid privacy and revelation of information by attempting to minimize the impact on the sanitized transactions.

Problem statement: Specification of an Panglossian Solitary-Skim Sanitization for Privacy Preserving Data Archeology algorithm based technique and to design, develop and implement functionalities like User friendly framework, secure protocol for preserving private data's and knowledge, Reusability, Portability. We have adapted the one-scan SWA christening it as Panglossian Solitary-Skim Sanitization algorithm and used it for Privacy Preserving Data Archaeology. We have utilized the notion of disclosure threshold for each lone pattern to curb and provide an enhanced agility allowing an administrator to place disparate weights for varied rules.

Problem description: Data Archeology is an important technique for exploratory data analysis and useful in many practical domains such as Health information, Financial information, Genetic information, Criminal justice and Location information. Privacy Preserving Data Archeology is merging the data of the users without disclosing the private and sensitive details of the users.

We have to develop mechanism for modifying the unique facts by some means, with the intention that the private data and private knowledge linger private even subsequent to the mining process. There are many mechanisms which have been adopted for privacy preserving data mining. Owing to the versatility of the data mining tasks, a family of Privacy-Preserving Data Amendment/alteration (PPDA) methods for protecting privacy before data are shared can be used to the address privacy preservation in data mining. Lots of Portrayal of the efficacy of privacy-preserving mechanism up till now, have all focused on proving the viability of privacy-preserving calculation, excluding the advantages/ disadvantages of selection of such mechanisms. But the method used here is one which removes highest frequency items of occurrences from the original database. This is done by the process name sanitization (i.e.) transforming a transactional database to be shared in such a way that the restrictive rules cannot be discovered. The effectiveness of the data sanitization is measured by the proportion of restrictive rules effectively hidden (hiding failure), the proportion of rules accidentally hidden (misses cost) and the amount of artifactual rules created by the process. The sanitization process acts on the data to remove or hide a group of restrictive association rules that contain sensitive knowledge.

ARCHITECTURE OF THE PROPOSED WORK

We bring out a diagrammatic schematic representation of the blocks as shown in Fig. 2 involved in the proposed architecture.

Block diagram: We have specified an panglossian conceptual framework as shown in Fig. 3 in order to effectively use this technique in a general pedestal which will be the basis for ascertaining the Panglossian solitary-skim sanitization technique for a given type of application.

Original database as shown in Fig. 3 may be a depiction of a database serveror data warehouse server. These datasets and rules may be owned either by a single party or by various parties who are in all probability forbidden from partaking, or not agreeable to dole, their datasets. In this block data sets are given as an input, which in the course of action gets converted to a uniform format as shown in Fig 1 which is prescribed by the PPDM framework designer. It contains data of any nature viz., relational and complex types of data; heterogeneous/ transactional/objectoriented/objectrelational/temporal/ time-series/text/multimedia/legacy databases and global information systems (WWW).

Data preprocessing: It is done in the real world as these data are dirty viz., incomplete (lacking attribute values, lacking certain attributes of interest, or containing only aggregate data), noisy (containing errors or outliers), inconsistent: containing discrepancies in codes or names.

Data cleaning: During Data cleaning as shown in Fig 4. we fill in missing values, smooth noisy data, identify or remove outliers and resolve inconsistencies by correcting inconsistent data.

Data integration: It is a process as shown in Fig. 5 to coalesce data from manifold sources (multiple databases, data cubes, or files) into coherent store. Schema integration is done to integrate metadata from different sources. Detecting and resolving data value conflicts is a necessity for the same real world entity, since attribute values from heterogeneous data sources are different because of their dissimilar representations, different scales followed worldwide.

Data transformation: The Data transformation is performed by any one of the following: smoothing (removing noise from data), Normalization (scaled to fall within a small, specified range by min-max normalization-score normalization, normalization by decimal scaling), Attribu/feature construction (New attributes constructed from the given ones), Generalization (concept hierarchy climbing) and aggregation (summarization, data cube

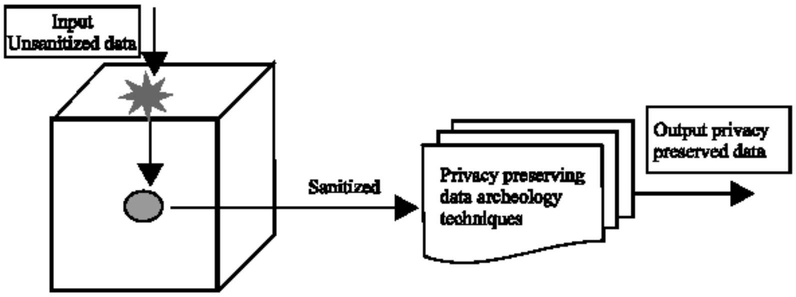


Fig. 2: Sanitization of panglossian solitary-skim sanitization for privacy preserving data archeology: high-level

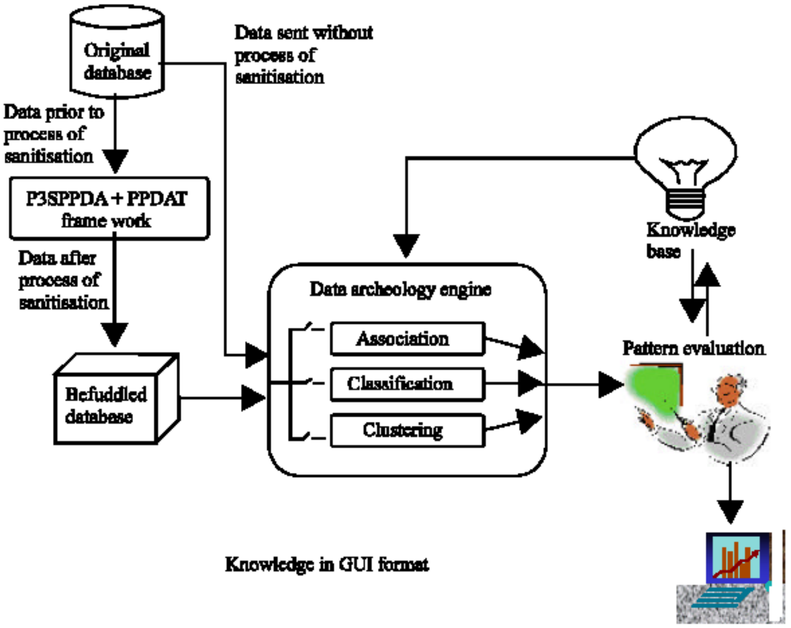


Fig 3: Block diagram of the proposed architecture

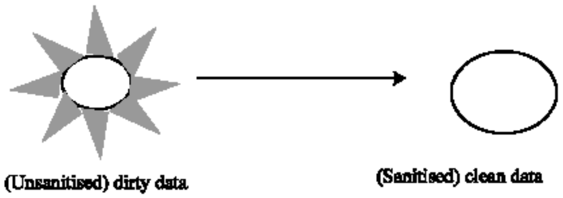


Fig. 4: Cleaning of data

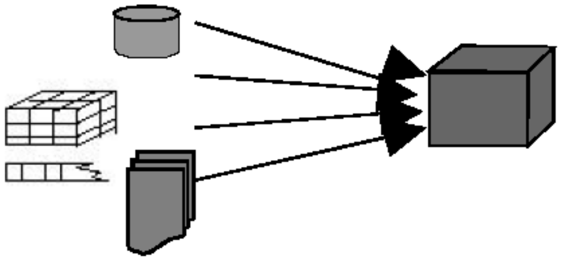


Fig 5: Data integration

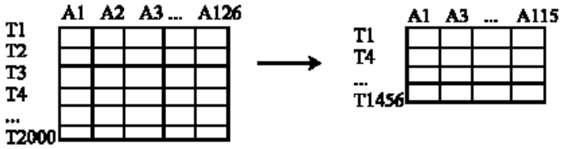


Fig. 6: Data reduction

construction). Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results. The Data reduction strategies used are Data cube aggregation, Dimensionality reduction, Numerosity reduction, Discretization and concept hierarchy generation.

Feature extraction: obtaining only the interesting attributes of the data, e.g., date acquired is probably not useful for clustering celestial objects, as in Skycat.

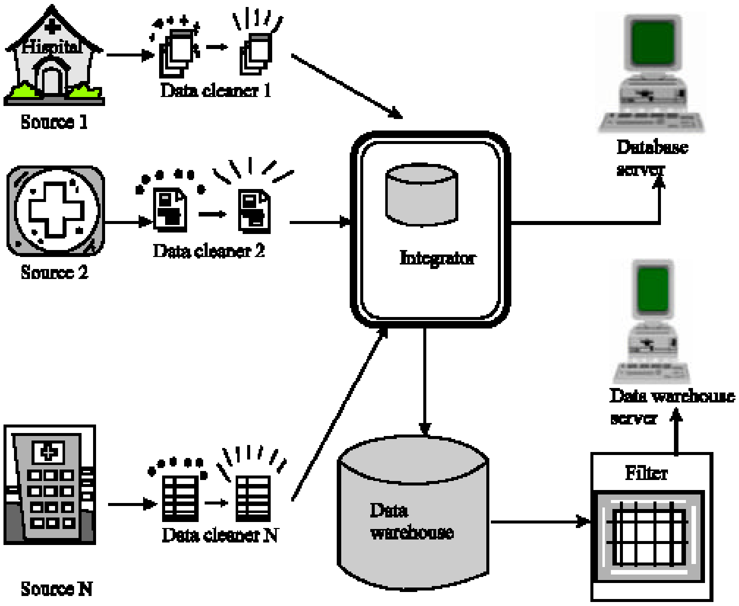


Fig. 7: Data collection and storage

Pattern extraction and discovery with the help of knowledge bases: In this block we extract the interesting (non-trivial, implicit, previously unknown and potentially useful) information or patterns from data in large databases.

Visualization of the data: The principal graphical techniques used for the visually representing the mined knowledge for further analysis of the information viz., Box plot, Histogram, Multi Vari chart, Run chart, Pareto chart, Scatter plot, Stem- and -leaf plot, tree diagram, bar chart, pie chart, function graph, scatter plot, Euler diagram, Venn diagram, existential graph etc.

Evaluation of results: Based on the visual representations of data we can do decision-making.

Database server: If it is a database server as shown in Fig 7 Data Collection and storage it stores current, up-to-date detailed, flat relational isolated data's after the process of cleaning and integrating from the different databases. It contains data's which are current, up-to-dtae detailed, flat relational and isolated. The storage capacity of a database server is usually 100MB-GB.

Data warehouse server: If it is a data warehouse server as shown in Fig. 7 then it is a subject-oriented, integrated, time-variant and nonvolatile collection of data in support of management's decision-making process (W. H. Inmon).

From different data warehouses the data are filtered for the required domain and stored in the data warehouse server. It contains data's which are Historical, consolidated, summarized, multidimensional data's. The storage capacity of a data warehouse server is usually 100 GB-TB.

SYSTEM ARCHITECTURE DESIGN

Datasets used: We have tested our algorithm on a real-time database collected from various hospitals in Chennai. Among this data we have chosen the accidents data which is stored in the database.

GUI: The graphical user interface is designed as shown in Fig. 8.

GUI description:

- ComboBox1: User Threshold : User can either select 50% -partial hiding or 100% -full hiding.
- ComboBox1: Support: User can select which of the items are sanitized and their corresponding support.
- Text Filed: Datawarehouse name: The name of the datawarehouse that is the sanitized.
- Button1: Desanitized Database: When the user clicks this button the database before sanitization is displayed as shown in the Fig. 9.

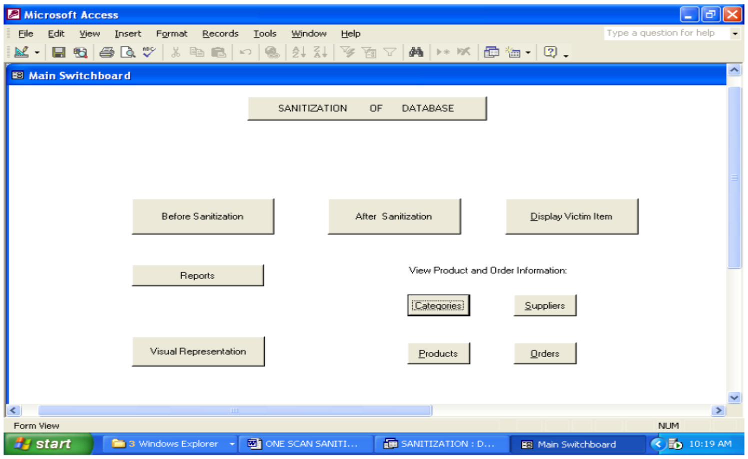
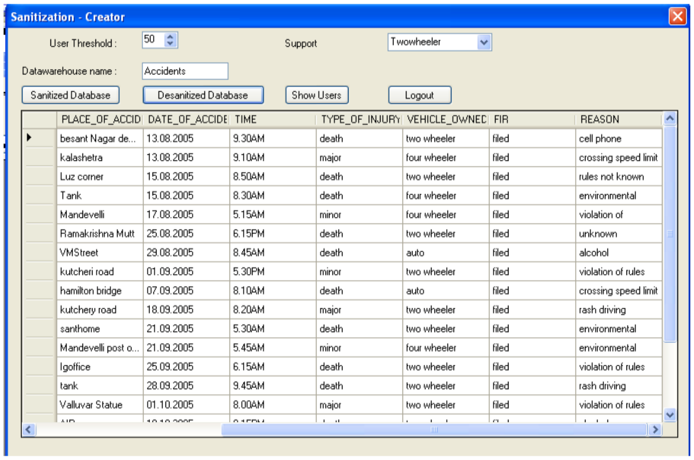


Fig. 8: Graphical user interface



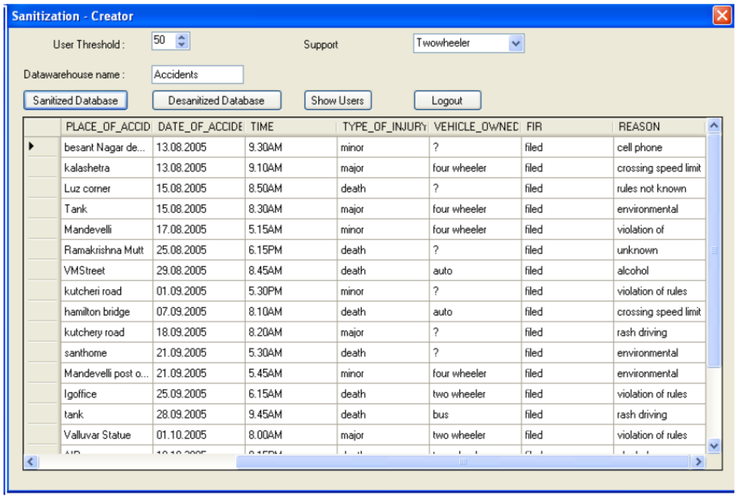
User Threshold : 50

Support: Twowheeler

Datawarehouse name : Accidents

PLACE_OF_ACCID	DATE_OF_ACCIDE	TIME	TYPE_OF_INJURY	VEHICLE_OWNEC	FIR	REASON
besant Nagar de...	13.08.2005	9.30AM	death	two wheeler	filed	cell phone
kalashetra	13.08.2005	9.10AM	major	four wheeler	filed	crossing speed limit
Luz corner	15.08.2005	8.50AM	death	two wheeler	filed	rules not known
Tank	15.08.2005	8.30AM	death	four wheeler	filed	environmental
Mandevelli	17.08.2005	5.15AM	minor	four wheeler	filed	violation of
Ramakrishna Mutt	25.08.2005	6.15PM	death	two wheeler	filed	unknown
VMStreet	29.08.2005	8.45AM	death	auto	filed	alcohol
kutcheri road	01.09.2005	5.30PM	minor	two wheeler	filed	violation of rules
hamilton bridge	07.09.2005	8.10AM	death	auto	filed	crossing speed limit
kutchery road	18.09.2005	8.20AM	major	two wheeler	filed	rash driving
santhome	21.09.2005	5.30AM	death	two wheeler	filed	environmental
Mandevelli post o...	21.09.2005	5.45AM	minor	four wheeler	filed	environmental
Igoffice	25.09.2005	6.15AM	death	two wheeler	filed	violation of rules
tank	28.09.2005	9.45AM	death	two wheeler	filed	rash driving
Valluvar Statue	01.10.2005	8.00AM	major	two wheeler	filed	violation of rules

Fig. 9: Display of data before sanitization



User Threshold : 50

Support: Twowheeler

Datawarehouse name : Accidents

PLACE_OF_ACCID	DATE_OF_ACCIDE	TIME	TYPE_OF_INJURY	VEHICLE_OWNEC	FIR	REASON
besant Nagar de...	13.08.2005	9.30AM	minor	?	filed	cell phone
kalashetra	13.08.2005	9.10AM	major	four wheeler	filed	crossing speed limit
Luz corner	15.08.2005	8.50AM	death	?	filed	rules not known
Tank	15.08.2005	8.30AM	major	four wheeler	filed	environmental
Mandevelli	17.08.2005	5.15AM	minor	four wheeler	filed	violation of
Ramakrishna Mutt	25.08.2005	6.15PM	death	?	filed	unknown
VMStreet	29.08.2005	8.45AM	death	auto	filed	alcohol
kutcheri road	01.09.2005	5.30PM	minor	?	filed	violation of rules
hamilton bridge	07.09.2005	8.10AM	death	auto	filed	crossing speed limit
kutchery road	18.09.2005	8.20AM	major	?	filed	rash driving
santhome	21.09.2005	5.30AM	death	?	filed	environmental
Mandevelli post o...	21.09.2005	5.45AM	minor	four wheeler	filed	environmental
Igoffice	25.09.2005	6.15AM	death	two wheeler	filed	violation of rules
tank	28.09.2005	9.45AM	death	bus	filed	rash driving
Valluvar Statue	01.10.2005	8.00AM	major	two wheeler	filed	violation of rules

Fig. 10: Display of data after sanitization

- Button 2: Sanitized Database: When the user clicks this button the database after Sanitization is displayed as shown in the Fig. 10.
- User Form: This form has a login page which has got validation for the users.

IMPLEMENTAION OF SANITIZATION USING SANITIZING ALGORITHMS

Different sanitizing algorithms:

- Sliding Window Algorithm (SWA).
- Algo2a hides restrictive rules by reducing support.
- Item Grouping Algorithm (IGA) groups restricted association rules in clusters of rules sharing the same itemsets. The shared items are removed to reduce the impact on the sanitized dataset.
- A sanitizing algorithm called Downright Sanitizing Algorithm (DSA)sanitizes a set of restrictive rules while blocking some inference channels.
- Naïve Algorithm, sanitizes restrictive itemsets and their supersets.

Panglossian solitary-skim sanitization algorithm: The Sliding Window Algorithm (SWA) improves the poise between privacy and knowledge discovery in association rule mining. This algorithm requires only one pass over a transactional database despite the consequences of the database size and the number of restrictive association rules that must be cosseted. Sliding Window Algorithm (SWA) scans a group of K transactions, at a time and sanitizes the restrictive rules present in such transactions based on a disclosure threshold D_T defined by a database owner. We set the window size of SWA to 20000 transactions in both datasets. In this research, we have adapted the one-scan SWA christening it as Panglossian Solitary-Skim Sanitization algorithm and used it for Privacy Preserving Data Archaeology. We have utilized the notion of disclosure threshold for each lone pattern to curb and provide an enhanced agility allowing an administrator to place disparate weights for varied rules.

Steps of the Algorithm: Step 1: Identify and Pigeonhole transaction as susceptible operation (if it contains any restrictive rules) and non-susceptible ones.

For each transaction read from a database D, we identify if this transaction contains any restrictive association rules. If not, the transaction is copied directly to the sanitized database D0. Otherwise, this transaction is sensitive and must be sanitized.

Step 2: Select the victim item. A transaction may contain lot of items. The item with the highest frequency count is the victim item. In some cases the victim item is selected randomly.

Step 3: The threshold is given. The threshold is given by the database owner for sanitizing the sensitive transaction. The reason for why the threshold is taken into consideration is that when one company is giving the database to the other if the company sanitizes all the sensitive transaction, then it is of no use to the supplier, instead there should be balance between providing reasonable information at the same time privacy preserving.

Step 4: Sorting the sensitive transactions by size. In this step, we sort the number of sensitive transactions computed in the previous step, for each restrictive rule. The sensitive transactions are sorted in ascending order of size. Thus, we start marking the shortest transactions to be sanitized. By removing items of shortest transactions we will be minimizing the impact on the sanitized database since shortest transactions have less combinations of association rules.

Step 5: Sanitizing the sensitive transaction. Removal of the victim item, from the sensitive transaction is done in this step. Every time we remove a victim item from a sensitive transaction, we perform a look ahead procedure to verify if that transaction has been selected as a sensitive transaction for other restrictive rules.

If so and the victim item we just removed from the current transaction is also part of this restrictive rule, we remove that transaction from the list of transaction ID's marked in the other rules. In doing so, the transaction will be sanitized and then copied to the sanitized transaction database.

SWA has essentially five steps. In the first, the algorithm sorts the items of each transaction t, in ascending order, to identify if the transaction is sensitive or not by using a binary search fashion. A transaction is sensitive if it contains all items of at least one restrictive rule. In this case, the transaction ID is added to the list of transition Ids of the corresponding restrictive rule. Then, the algorithm computes the frequencies of the items of the restrictive rules that are present in such a transaction.

Sanitizing the shortest transactions in each restrictive rule minimizes the impact on the sanitized database because shortest transactions contain less combinations of association rules. In step 5, the sensitive transactions to be cleansed are first marked and then sanitized. If the disclosure thres hold is 0 (i.e. all restrictive rules need to be hidden), we do a look ahead in the Mining Permissions (MP) to check whether a sensitive transaction need not be sanitized more than once. This is to improve the misses cost. The function look ahead () looks in MP from rri onward whether a given transaction t is selected as sensitive transaction for another restrictive rule r. If it is the case and Victim rri is part of the restrictive rule r, the transaction t is removed from the list since it has just been sanitized already.

RESULTS AND DISCUSSION

Figure 11a and b shows that up to 3000 transactions the difference between the original and the sanitized difference remains the same. Where D is the database improves slightly. After 3000 transactions, the transaction database before sanitization and D' is the sanitized database.

The one scan sanitization is implemented using visual basic why this was choosen is because its user friendly and the databases are stored using Microsoft access. Here is a sample database (Fig. 12). The experimental results show that in the database the highest frequency of occurring item such jam has been found out using the support count of the transaction database. Then the particular item is removed an the sanitized database is shown in the database below (Fig. 13). These sanitized database of a company plays an important role in their collaborative dealings.

Evaluation parameters: Performance time taken to sanitize the database in milli seconds. And this process of santization may function effectively when the item set is moderate.

Scalability: The algorithm is an optimal one for medium datasets but when the interestingness measures increasing the algorithm may not be feasible.

The victim data item changes dynamically. The following line graph displays their support change before sanitization and after sanitization (Fig. 14a, b).

Reliability: This parameter is a non functional parameter.It is defined as obtaining the correct output with in the given time.The reliability with respect to this application is determined to be getting the right support or confidence value with in the estimated time (Fig. 15).

Availability: This is defined as the availability of the data for the application to sanitize. For sanitizing vital is the data. If the data is not made available the application cannot perform its function (Fig. 16).

Response time: This is defined as the time taken by the application to sanitize the data once the user submits the data for sanitization (Fig. 17).

Scalability: This is defined as the capability of the application to scale up from small data to larger data for the process of sanitization (Fig. 18).

Performance: This parameter is obtained by calculating the weighted sum of all the above non functional parameters. Reliability, Availabiltiy, Scalability needs to be high and Response Time needs to be very low (Fig. 19).

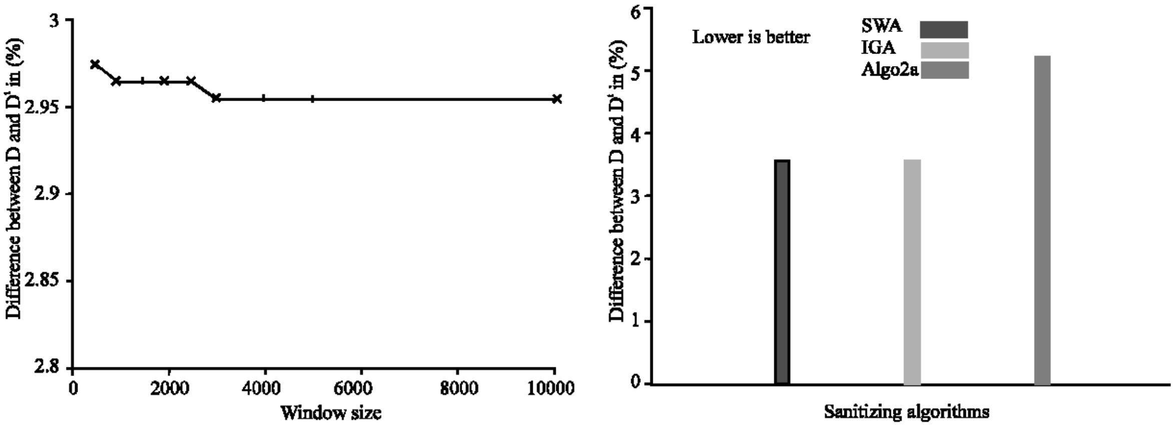
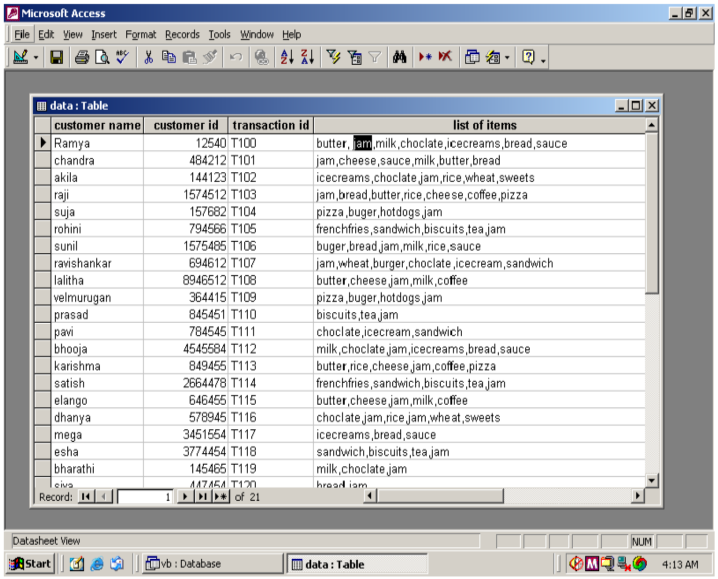
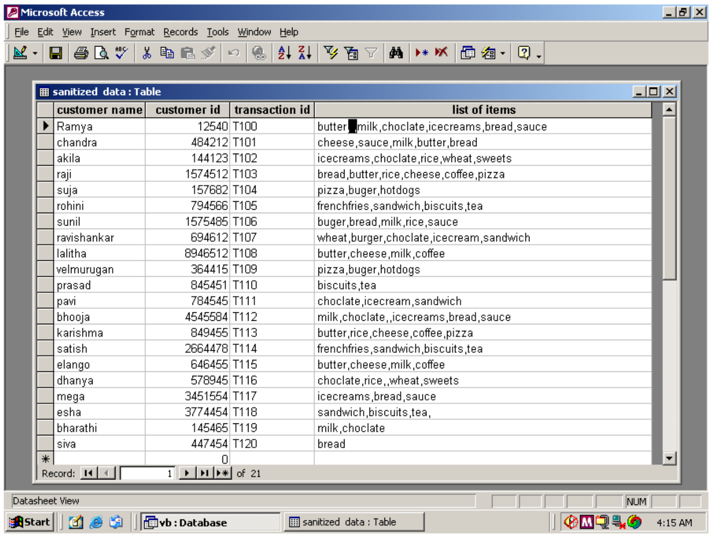


Fig. 11a, b: A representation of comparison between the sanitizing algorithm is shown



customer name	customer id	transaction id	list of items
Ramya	12540	T100	butter, jam,milk,choclate,icecreams,bread,sauce
chandra	484212	T101	jam,cheese,sauce,milk,butter,bread
akila	144123	T102	icecreams,choclate,jam,rice,wheat,sweets
raji	1574512	T103	jam,bread,butter,rice,cheese,coffee,pizza
suja	157682	T104	pizza,buger,hotdogs,jam
rohini	794566	T105	frenchfries,sandwich,biscuits,tea,jam
sunil	1575485	T106	buger,bread,jam,milk,rice,sauce
ravishankar	694612	T107	jam,wheat,burger,choclate,icecream,sandwich
lalitha	8946512	T108	butter,cheese,jam,milk,coffee
velmurugan	364415	T109	pizza,buger,hotdogs,jam
prasad	845451	T110	biscuits,tea,jam
pavi	784545	T111	choclate,icecream,sandwich
bhooja	4545584	T112	milk,choclate,jam,icecreams,bread,sauce
karishma	849455	T113	butter,rice,cheese,jam,coffee,pizza
satish	2664478	T114	frenchfries,sandwich,biscuits,tea,jam
elango	646455	T115	butter,cheese,jam,milk,coffee
dhanya	578945	T116	choclate,jam,rice,jam,wheat,sweets
mega	3451554	T117	icecreams,bread,sauce
esha	3774454	T118	sandwich,biscuits,tea,jam
bharathi	145465	T119	milk,choclate,jam
siva	447454	T120	bread,jam

Fig 12: Database before sanitization using sliding window algorithm



customer name	customer id	transaction id	list of items
Ramya	12540	T100	butter,milk,choclate,icecreams,bread,sauce
chandra	484212	T101	cheese,sauce,milk,butter,bread
akila	144123	T102	icecreams,choclate,rice,wheat,sweets
raji	1574512	T103	bread,butter,rice,cheese,coffee,pizza
suja	157682	T104	pizza,buger,hotdogs
rohini	794566	T105	frenchfries,sandwich,biscuits,tea
sunil	1575485	T106	buger,bread,milk,rice,sauce
ravishankar	694612	T107	wheat,burger,choclate,icecream,sandwich
lalitha	8946512	T108	butter,cheese,milk,coffee
velmurugan	364415	T109	pizza,buger,hotdogs
prasad	845451	T110	biscuits,tea
pavi	784545	T111	choclate,icecream,sandwich
bhooja	4545584	T112	milk,choclate,,icecreams,bread,sauce
karishma	849455	T113	butter,rice,cheese,coffee,pizza
satish	2664478	T114	frenchfries,sandwich,biscuits,tea
elango	646455	T115	butter,cheese,milk,coffee
dhanya	578945	T116	choclate,rice,,wheat,sweets
mega	3451554	T117	icecreams,bread,sauce
esha	3774454	T118	sandwich,biscuits,tea,
bharathi	145465	T119	milk,choclate
siva	447454	T120	bread
*		0	

Fig. 13: Database after sanitization using sliding window algorithm

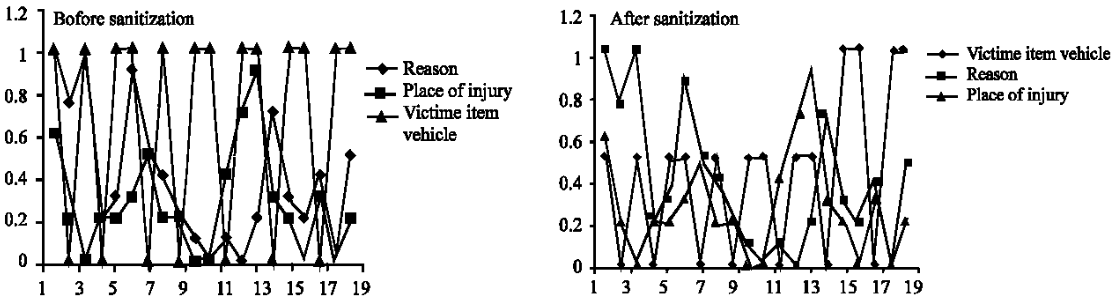


Fig. 14a, b: Graphs before and after sanitization

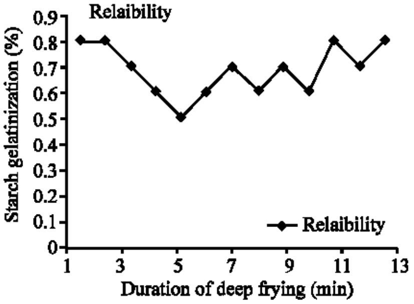


Fig. 15: Reliability is determined to be getting the right support or confidence value with in the estimated time

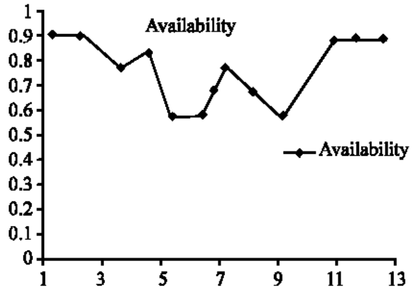


Fig. 16: The availability of the data

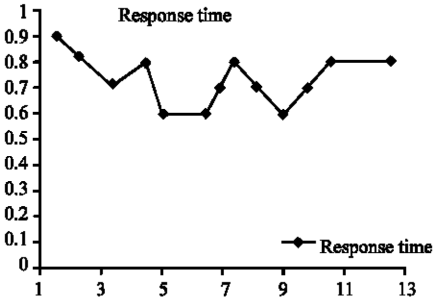


Fig. 17: The time taken by the application to sanitize the data

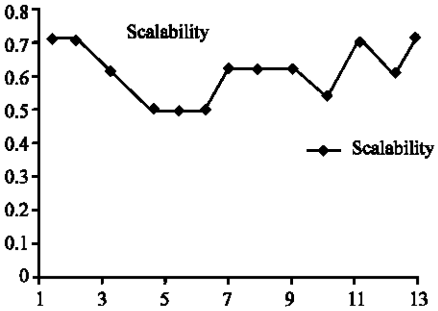


Fig. 18: The capability of the application to scale up from small data to larger data

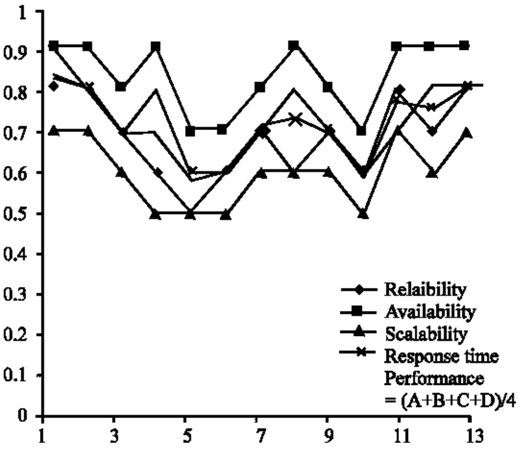


Fig. 19: The weighted sum of all the above non functional parameters

CONCLUSION

The concept of sanitization is an emerging one in domains such as the market domain which we have choosen. Based on the sanitized database strategic decision are being made in the business world. So its important for the productivity and development of the organization.Thus the concept read in the study has been implemented using Visual Basic as front end and Microsoft Access as back end.

FUTURE WORK

The study is implemented for removing a particular item which has the highest frequency of occurrences. Items occurring on the next order of frequency can also be removed but the threshold value has to be taken into consideration.

Privacy and efficiency are the two eyes of a person, equally imperative for safe Information Harvesting. Compromising on both is not sensible. Hence we have suggested a framework for Privacy Preserving Information Harvesting. We need to implement and evaluate true efficiency, after including improvements such as sampling.

Future research will attempt to demonstrate the viability of the architecture through a proof-of-concept prototype. We demonstrate how perturbation technique can be effectively done using this architecture. In the future, we hope to perk up the efficiency of this approach. As a first direction, we sketch to investigate firmly generating diplomat samples from the database. This would be an orthogonal technique for applications not requiring perfect accuracy, but very high security. We hope the proposed solution will get hold of new frameworks, techniques, paving way for research track and work well according to the evaluation metrics including hiding effects, data utility and time performance.

REFERENCES

Atallah, M., E. Bertino, A. Elmagarmid, M. Ibrahim and V. Verykios, 1999. Disclosure Limitation of Sensitive Rules. In: Proc. IEEE. Knowledge and Data Engineering Workshop, Chicago, Illinois, pp: 45-52.

Berry, M. and G. Linoff, 1997. Data Mining Techniques for Marketing, Sales and Customer Support. John Wiley and Sons, New York, USA.

Castano, S., M. Fugini, G. Martella and P. Samarati, 1995. Database Security. Addison-Wesley Longman Limited, England.

Clifton, C. and D. Marks, 1996. Security and Privacy Implications of Data Mining. In Workshop on Data Mining and Knowledge Discovery, Montreal, Canada, pp: 15-19.

Dasseni, E., V.S. Verykios, A.K. Elmagarmid and E. Bertino, 2001. Hiding Association Rules by Using Confidence and Support. In: Proc. 4th Inform. Hiding Workshop, Pittsburg, PA., pp: 369-383.

Johnsten, T. and V.V. Raghavan, 1999. Impact of Decision-Region Based Classification Mining Algorithms on Database Security. In: Proc. 13th Annual IFIP WG 11.3 Working Conference on Database Security, Seattle, USA., pp: 177-191.

Oliveira, S.R.M. and O.R. Zaian, 2003. Algorithms for Balancing Privacy and Knowledge Discovery in Association Rule Mining. In: Proc. 7th International Database Engineering and Applications Symposium (IDEAS), Hong Kong, China.

Oliveira, S.R.M. and O.R. Zaian, 2002. Privacy Preserving Frequent Itemset Mining. In: Proc. IEEE. ICDM. Workshop on Privacy, Security and Data Mining, Maebashi City, Japan, pp: 43-54.

Oliveira, S.R.M. and O.R. Zaian, 2003. An Efficient One-Scan Sanitization For Improving The balance between privacy and knowledge Discovery. Technical Report, TR 03-15.

Saygin, Y., V.S. Verykios and C. Clifton, 2001. Using Unknowns to Prevent Discovery of Association Rules. SIGMOD. Rec., 30: 45-54.