Protecting Sensitive Association Rules in Privacy Preserving Data Mining using Genetic Algorithms

S.Narmadha
Research Scholar
Department of Computer Science
Bharathiar University
Coimbatore.

S.Vijayarani
Assistant Professor
Department of Computer Science
Bharathiar University
Coimbatore.

ABSTRACT
Privacy and security risks taken place from the application of different data mining techniques to great organizational data storehouses have been exclusively inspected by a new research domain, the so-called privacy preserving data mining. Association rule hiding is one of the privacy preserving techniques which study the problem of hiding sensitive association rules. There are many algorithms and techniques were developed to solve this problem. In this research work, we have used genetic algorithm optimization technique for protecting sensitive association rules.

Keywords
Privacy, Association rule, Sensitive, Hiding, Genetic Algorithm.

1. INTRODUCTION
Privacy preserving data mining is a novel research area that examines the troubles which occurs after applying the data mining techniques. Privacy problems related to the application of data mining techniques are divided into two broad kinds, data hiding and knowledge hiding. Data hiding is the exclusion of confidential or sensitive information from the data before it is disclosed to others. Knowledge hiding is the results of data mining techniques, after having analyzed the data, these may find out the hidden knowledge. Such knowledge should be protected from others.

Privacy offers emancipation from illegal entrance. The long term goal of the government statistical agencies and database security research community is how to secure the sensitive data against unconstitutional access. Privacy protection has become one of the major issues in data mining research. An essential constraint of privacy-preserving data mining is to safeguard the input data, yet still permit data miners to dig out the valuable knowledge models. Many numbers of privacy-preserving data mining techniques have newly been projected which take either a cryptographic or a statistical approach. Secure multi-party computation is used in the cryptographic approach which ensures strong privacy and accuracy. But, this approach typically suffers from its poor performance. The statistical approach has been used to extract the facts from decision trees, association rules, and clustering. This approach is very popular because of its high performance.

The association rule hiding problem can be considered as a deviation of the well identified database inference control problem in statistical and multilevel databases. The primary goal in database inference control is to guard access to sensitive information that can be obtained through non sensitive data and inference rules. In association rule hiding, we think about that it is not the data itself but somewhat the sensitive association rules that produce a breach to privacy. Given a set of association rules, which can be extracted from a specific data collection and are considered to be sensitive, the task of association rule hiding algorithms is to properly transform the original data so that the association rule mining algorithms that may be applied to this modified data (i) will be incompetent to determine the sensitive rules (ii) will be capable to mine all the non sensitive rules that become visible in the original dataset and (iii) will be incapable to discover false rules.[1]

The technique developed in this paper uses binary transactional dataset as a contribution and modifies the original dataset based on the idea of genetic algorithms in such a way that all the susceptible rules are concealed without any loss of data. The most promising approach for transaction modification is the alteration of original database (i.e., by replacing 1's by 0's and vice versa). The modification process can influence the original set of rules, that can be mined from the original database, either by hiding rules which are not sensitive (*lost rules*), or by introducing rules in the mining of the modified database, which were not supported by the original database (*ghost rules*). [1]

The rest of the paper is organized as follows. In Section 2, the related works are discussed. In Section 3, the general problem formulation and the basic definitions of association rule mining are given. In Section 4, we present the algorithm which implements the distortion technique based on genetic algorithm approach. In Section 5 the effectiveness of the algorithm can be evaluated and we have presented the experimental results of the proposed technique. We conclude the paper in Section 6.

2. RELATED WORK
There are three types of association rule hiding algorithms namely heuristic approaches, border-based approaches and exact approaches. A heuristic approach involves regimented, fast algorithms that carefully cleanse a set of transactions from the database to hide the sensitive knowledge. Due to their effectiveness and scalability, the heuristic approaches have been the focus of consideration for the huge majority of researchers in the knowledge hiding field. Border based approaches considers the process of sensitive rule hiding through modification of the original borders in the lattice of the numerous and the intermittent patterns in the dataset. In these schemes, the sensitive knowledge is hidden by enforcing the revised borders in the sanitized database. Exact approaches are non-heuristic algorithms which envisage the hiding process as a constraint satisfaction problem that they solve using integer or linear programming. [3]

Yucel Saygın, Vassilios S.Verkios, Ahmed K. Elmagarmid [14] using the "unknown symbol" to keep the privacy in data mining. First, they introduced a new symbol in the alphabet of an item. The possible set of values of an item in the new setting becomes { 0, 1, ?}. For example, the value in the i^{th} position of a transaction is 1 if the transaction contains (or supports) the i^{th} item and, the value is 0 otherwise. A "?" mark in the i^{th} position of a transaction means that do not have any information regarding whether the transaction contains the i^{th} item or not. With the new approach that involves "?" marks, the definition of support should be modified.

V.S. Verykios, Emmanuel D. Pontikakis, Yannis Theodoridis, Liwu Chang [13] provided two fundamental approaches in order to protect sensitive rules from disclosure. The first approach prevents rules from being generated, by hiding the frequent sets from which they are derived. The second approach reduces the significance of the rules by setting their confidence below a user-specified threshold. Five algorithms are used for hiding the sensitive association rules based on these two approaches. The first three algorithms are rule-oriented. In other words, they decrease either the confidence or the support of a set of sensitive rules, until the rules are hidden. This can happen either because the large item sets that are associated with the rules are becoming small or because the rule confidence goes below the threshold. The last two algorithms are item set-oriented. They decrease the support of a set of large item sets until it is below a user-specified threshold, so that no rules can be derived from the selected item sets.

Nan Zhang, Shengquan Wang, and Wei Zhao [9] have introduced a new algebraic technique. In this technique have identified the association rules more precisely and disclose a lesser amount of private information. In comparison with previous approaches, this method has introduced a two-way communication mechanism between the data miner and data providers with little overhead. In particular, the data miner sends perturbation guidance to the data providers. Using this intelligence, the data providers disfigure the data transactions to be transmitted to the miner. As a result, new scheme identifies association rules more precisely than previous approaches and at the same time reaches a higher level of privacy.

Mohammad Naderi Dehkordi [8] introduced new multi-objective method for hiding sensitive association rules based on the concept of genetic algorithms. The main purpose of this method is fully supporting security of database and keeping the utility and certainty of mined rules at highest level. In their work, they have used four sanitization strategies such as confidence, support, hybrid and max-min. They introduced the idea of both rule and item set sanitization, which complements the old idea behind data sanitization.

3. PROBLEM FORMULATION

Consider a Database D consists of set of transactions $D=\{T_1,T_2, \ldots, T_n\}$ and each transaction consists of set of items $I=\{I_1,I_2,\ldots,I_m\}$ where $T_i=\{I_{i1},I_{i2}, \ldots, I_{ik}\}$ and $I_{ij} \in$ I, the Association Rule Problem is to identify all association rules X $\Rightarrow$ Y with a minimum support and confidence. There are two techniques in association rule mining with frequent item set namely, one is to find large item sets and another one is generate rules from frequent item sets. In this work, we have used apriori algorithm for finding frequent item sets in a large database.

Let $I= \{I_1 \ldots I_2 \ldots I_m\}$ be a set of items and let *D* is the dataset of transactions that the goal of sanitization is its alteration in order to no sensitive rule disclosed[5]. Any $X \in I$ is an item set. Each item set which contains *k* items called k-item sets. Let $T= \{T_1, T_2 \ldots T_n\}$ be a set of transactions. The well known measure in frequent item set mining is support of item set. The support measure of an item $X \in I$ in database *D*, is the count of transactions contain *X* and denoted as Support count(X). An item set *X* has support measure *s* in dataset D if *s%* of transactions support *X* in dataset *D*. Support measure of *X* is denoted as Support(X).

$$\text{Support (X)} = \frac{\boldsymbol{Support\ Count(X)*100}}{\boldsymbol{n}}$$

Where n is the number if transaction in D. Item set *X* is frequent item set when Support(X) > SUP_{min} Where SUP_{min} is Minimum Support Threshold, which is predefined threshold. After mining frequent item sets, the association rule is a proposition of the form $X \rightarrow Y$, where $X, Y \in I$ and $X \cap Y = \emptyset$. The Confidence measure for rule $X \rightarrow Y$ in dataset *D* is defined

$$\text{Confidence (X}\rightarrow \text{Y)} = \frac{\boldsymbol{Support(XY)*100}}{\boldsymbol{Support(X)}}$$

The support is a measure of the frequency of a rule; the confidence is a measure of the strength of the relation between sets of items. Association rule mining algorithms scan the database of transactions and calculate the support and confidence of the candidate rules to determine if they are considerable or not. A rule is considerable if its support and confidence is higher than the user specified minimum support and minimum confidence threshold. In this way, algorithms do not retrieve all probable association rules that can be derivable from a dataset, but only a very small subset that satisfies the minimum support and minimum confidence requirements set by the users. An association rule-mining algorithm works as follows. It finds all the sets of items that appear frequently enough to be considered relevant and then it derives from them the association rules that are strong enough to be considered interesting. The major goal here is to preventing some of these rules that we refer to as "sensitive rules", from being revealed. The problem of privacy preserving in association rule mining (so called association rule hiding) focused on this paper can be formulated as follows. [7]

Consider a given transaction database D, minimum support threshold value SUP_{min}, minimum confidence threshold value $CONF_{min}$, a set of association rules AR can be mined from D and a set of sensitive association rules AR_{sen} can be mined from D. Sensitive association rules ($AR_{sen} \subseteq AR$) to be hidden then to generate a novel database D', such that the rules in $AR_{non\text{-}sen}=AR\text{-}AR_{sen}$ can mined from D' under the same threshold values SUP_{min} and $CONF_{min}$ C. After this process, the results of the solution are analyzed. In the performance analysis, we have to verify whether all the sensitive rules are hidden fully, non-sensitive rules are not affected (lost rules) and no extra fake rules are (ghost rules) erroneously will mined after the rule hiding process.

4. PROPOSED SOLUTION

In this section we will give the explanation of optimization technique. The following steps are required for the proposed solution.

Step 1: Consider a transactional database with a set of items and transactions

Step 2: Apriori algorithm is used to find the frequent item sets based on the minimum support threshold.

Step 3: From the frequent item sets, the set of association rules can be generated based on the minimum support and confidence thresholds.

Step 4: Select the sensitive rules from the set of association rules.

Step 5: Genetic algorithm is used for modifying the items based on the fitness function

Step 6: Repeat the steps 2 and 3 for the modified data set.

Step 7: Verify (i) all the sensitive rules are hidden, (ii) no non-sensitive rules are hidden (iii) no false rules

4.1 Apriori Algorithm

Apriori algorithm finds the regular set L in the database D. It makes utilize of the downward closure property. The algorithm is a bottom search, moving upward level; it prunes many of the sets which are unlikely to be frequent sets, thus saving any additional efforts.

4.1.1 Candidate Generation

Given the set of all frequent (k-1) item sets, we want to generate a superset of the set of all frequent k-item sets. The intuition behind the apriori candidate generation procedure is that if an item set X has minimum support, so do all subsets of X. after all the (l+1) candidate sequences have been generated, a new scan of the transactions is started and the support of these new candidates is determined[2].

4.1.2. Pruning

The pruning step removes the extensions of (k-1) item sets which are not found to be common, from being considered for counting support. For each transaction t, the algorithm checks which candidates are enclosed in t and after the last transaction are processed; those with support less than the minimum support are junked. [2].So we can discover the frequent item sets using the apriori algorithm and also generate rules from the frequent items.

4.1.3.. Apriori Algorithm

C_k : Candidate item set of size k

L_k : frequent item set of size k

L_1 = {frequent items};

C_{k+1} = candidates generated from L_k ;

for each transaction t in database do

increment the count of all candidates in

C_{k+1} that are contained in t

L_{k+1} = candidates in C_{k+1} with min_support

end

return $\cup_k L_k$

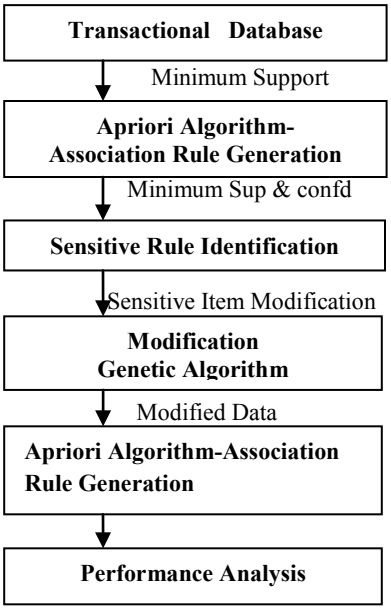

Figure 1. System Architecture

4.2. Optimization Technique

Optimization techniques are used for optimizing problems in which one needs to minimize or maximize a real function by methodically choosing the values of real or integer variables from within a particular set. It is finding the "best available" values of some objective function given a defined area, including a variety of different types of objective functions and different types of domains. Many types of optimization techniques and optimization algorithms are used in various types of approaches. In this paper we use the genetic algorithm for minimizing the cost function.

4.2.1. Genetic Algorithm

The genetic algorithm (GA) is an optimization and search technique based on the ethics of genetics and usual selection. 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).

In Genetic Algorithms, a population consists of a cluster of individuals called chromosomes that signify a complete solution to a certain problem. Each chromosome is a sequence of 0s or 1s. The initial set of the population is an erratically generated set of individuals. A new population is generated by two methods: steady state Genetic algorithm and generational Genetic Algorithm. The steady-state Genetic Algorithm replaces one or two members of the population; whereas the generational Genetic Algorithm replaces all of them at each generation of progression. In this work a generational Genetic Algorithm is adopted as population replacement method. In this method tried to keep a certain number of the best individuals from each generation and copies them to the new generation.

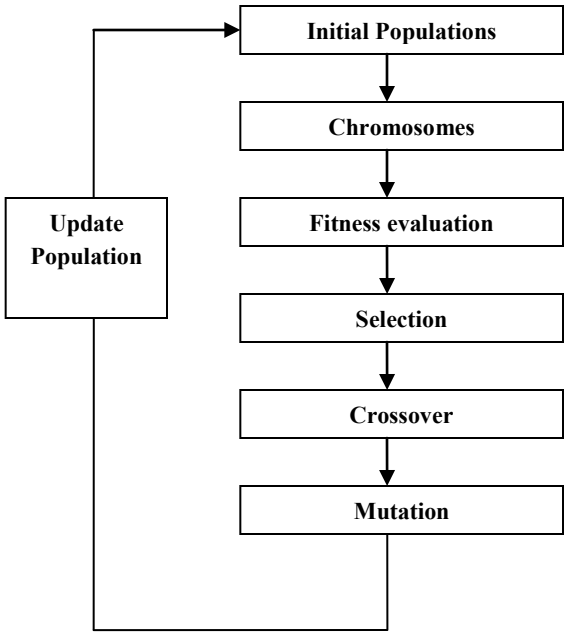

Figure 2. Genetic Algorithm

Each transaction is represented as a chromosome and occurrence of an i^{th} item in transaction showed by 1 and non occurrence of the item by 0 in i^{th} bit of transaction. The fitness of a chromosome is dogged by several methods and different strategies. Each population consists of several chromosomes and the best chromosome is used to generate the next population. For the initial population, a large number of random transactions are preferred. Based on the survival fitness, the population will make over into the future generation.

4.2.2. Fitness function

Fitness function is defined over the genetic representation and measures the superiority of the represented solution. The fitness function is forever problem dependent. Once we have the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly, and then improve it through repetitive application of mutation, crossover, and inversion and selection operators**.**

Fitness Function (f) = $\frac{\boldsymbol{Xi + Yi}}{\mathbf{2}}$

Where $X_i = \sum_{i=1}^{n} (\boldsymbol{itemset\ i = 1})$

This function is evaluated in each step of iteration.

4.2.3. Selection

In selection process, the individuals producing offspring are elected. The selection step is preceded by the fitness assignment which is based on the objective value. This fitness is used for the real selection process.

4.2.4. Crossover

Main function of crossover operation in genetic algorithms is blend two chromosomes mutually to generating novel offspring (child) [6]. Crossover occurs only with some probability (crossover probability). Chromosomes are not subjected to crossover remain unmodified. The perception following crossover is the exploration of new solutions and abuse of old solutions. Better fitness chromosomes have a prospect to be selected more than the inferior ones, so good solution always alive to the next generation. There are different crossover operators that have been developed for various purposes. Single point crossover and multi-point are the most famous operators. In this paper single-point crossover has been applied to make new offspring.

Table 1. Genetic Algorithm for modifying the sensitive items

1. {Initialization}
1.1 Initializing the sensitive items S_i, where
$S_i \in I$,
T= {t1, t2,……tn}
I={i_1,i_2,….i_n) , I ∈ T
1.2 Initializing the number of modifications required
Y_i=(S_i in T_i)
2. {Fitness function}
For each transaction T in D where T ∈ D
Do
For t=1 to n
Calculate cost (t) = (Xi + Yi)/2
n
Where $X_i = \Sigma$ (itemset$_i$=1)
i=1
Next
3. {Selection}
Choose t_i based on rank (cost (t))
Choose t_j based on rank (cost (t)) ≠ t_i
4. {Crossover}
Perform crossover (t_i, t_j)
Update items in t_i, t_j
5. {Mutation}
In Max(ST_i) and Max(Y_i) modify the item 1 as 0
Repeat the steps 1.3 to 4 until all the modification becomes 0
6. {Terminating}
Ensure all the sensitive items are modified
Number of modification becomes 0 then the process
Completed

7. Exit

4.2.5. Mutation

Mutation is a genetic operator that alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With these new gene values, the genetic algorithm may be able to arrive at better solution than was previously possible.

First the sensitive items and number of modifications required for each sensitive item are initialized. Next fitness function is evaluated for each transaction. Based on this fitness values, each transaction selection process are carried out in the third step. After the selection process, frequent items are updated through crossover operation. Crossover is the main process of genetic algorithm so in this step most of the frequent items become infrequent. Remaining items are modified in the mutation process. After ensuring the conditions i.e. all the sensitive items are modified then the process is completed and the execution is terminated. Finally apriori algorithm has been applied to the modified database for finding the frequent item sets for generating the sensitive rules. Now, we have to ensure that all the sensitive rules are hidden; no false rules are generated from the dataset and the non sensitive items are not affected.

5. EXPERIMENTAL RESULTS

Table 2. Sample database

a	b	c	d	e	f
1	0	1	1	0	0
1	1	1	0	1	0
0	1	1	0	1	0
0	1	0	1	0	0
1	1	1	0	0	1

The above table represents 5 transactions and 6 items. The presence of an item is represented as 1, and the absence of an item is 0.Threshold value such as support is 25% and confidence is 58%. From the above table items **a,b,c** are taken as a frequent item set using the apriori algorithm. Rules from the frequent item sets are as shown below, **Min sup=25% Min conf=58%**

Table 3. Association Rules

Association rules	Confidence
a→b	66.67%
a→c	100%
a→b,c	66.67%
a,b→c	100%
a,c→b	66.67%
b→c	75%
b,c→a	66.67%
b,c→e	66.67%
b,e→c	100%
c→a	75%
c→b	75%
c,e→b	100%
e→c	100%
e→b,c	100%
e→b	100%

Frequent item sets are used for generating association rules. Based on the threshold values, the sensitive rules are predicted. The items found in this rules are considered as sensitive items.

The transactions which contain the sensitive items are called population. The chromosomes of this population the fitness function has applied. After applying the crossover and mutation operations, based on fitness function the sensitive items of the original database are modified and for keeping the privacy of the database.

Table 4. Modified Database

a	b	c	d	e	f
1	0	0	1	0	0
1	1	0	0	1	0
0	0	1	0	1	0
0	1	0	1	0	0
0	0	1	0	0	1

After modification, apriori algorithm has been applied to verify all the sensitive rules are hidden with the same support and confidence. For analyzing the performance of genetic algorithm we have considered the following factors.

1. Hiding Failure
2. Sensitive Rule Protection
3. False rule generation
4. Misses cost or protection of Non sensitive rules
5. No. of iterations
6. Time complexity

5.1. Hiding Failure (HF)

This measure quantifies the percentage of the sensitive patterns that remain exposed in the sanitized dataset. It is defined as the fraction of the restrictive association rules that appear in the sanitized database divided by the ones that appeared in the original dataset. Formally,

$$HF = \frac{|ARsen(D')|}{|ARsen(D)|}$$

where $AR_{SEN}(D')|$corresponds to the sensitive rules discovered in the modified dataset D', $AR_{SEN}(D)|$ to the sensitive rules appearing in the original dataset D and |X| is the size of set X. Ideally, the hiding failure should be 0%.

We have applied these factors for different size of data sets with different thresholds. The sizes of the datasets are 1000 transactions and 50 items and 30 transactions and 20 items.

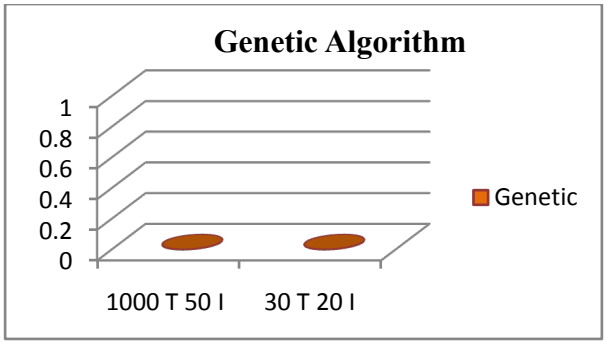

Figure 3. Hiding Failure

5.2. Sensitive Rule Protection

The figure 3 represents the hiding failure is 0% which means all the sensitive rules are protected from the disclosure. The accuracy of sensitive rule protection is 100%. It is represented in the figure 4.

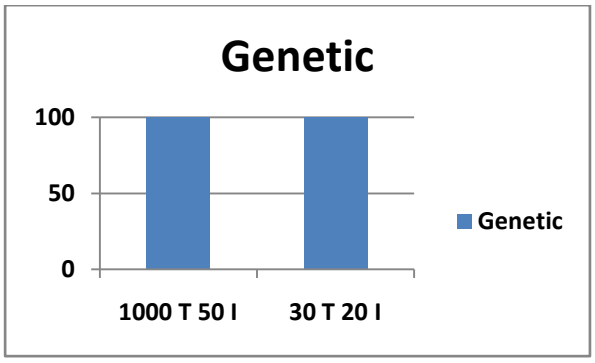

Figure 4. Protection of Sensitive Rules

The above chart represents genetic algorithm gives 100% accurate result for hiding the sensitive rules for two different data sets.

5.3. False Rule Generation

This measure computes the percentage of the false rules can be generated which is to be considered as a side-effect of the modification process. It is computed as follows:

$$FR = \frac{\pm[ARnon - sen(D) + ARsen(D)] - [ARnon - sen(D') + ARsen(D')]}{[ARnon - sen(D) + ARsen(D)]}$$

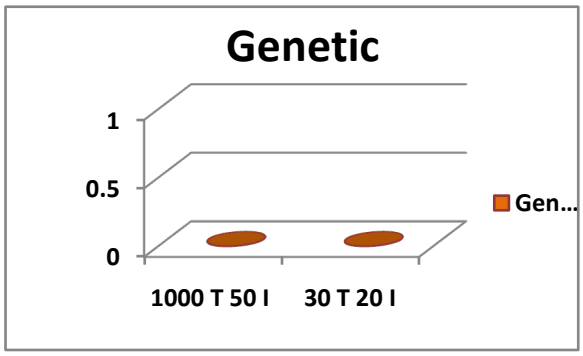

Figure 5. False Rules Generation

The above chart represents, in genetic algorithm, 0% of false rules are generated.

5.4. Misses Cost or Protection of Non-Sensitive Rules

This measure computes the percentage of the non-sensitive rules that are hidden as a side-effect of the modification process. It is computed as follows:

$$MC = \frac{|ARnon{-}sen(D){-}ARnon{-}sen(D')|}{|ARnon{-}sen(D)|}$$

where $AR_{NON-SEN}(D)$ is the set of all non-sensitive rules in the original database *D* and $AR_{NON-SEN}(D')$ is the set of all non-sensitive rules in the sanitized database *D'*.

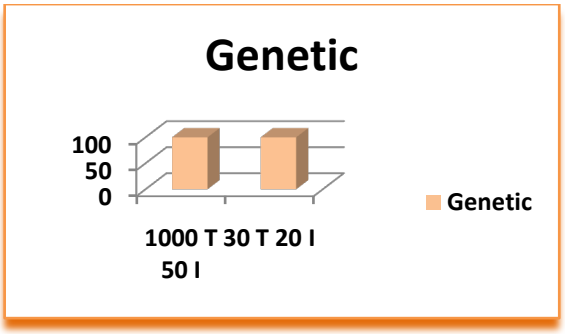

Figure 6. Protection of Non sensitive rules

The above chart represents during the modification process non sensitive rules are not affected. A genetic algorithm has given 100% protection for non sensitive rules.

5.5. No. of Iterations

Results of various threshold values are applied to get the frequent item sets and sensitive rules. The following chart shows the iterations required for genetic algorithm for different thresholds.

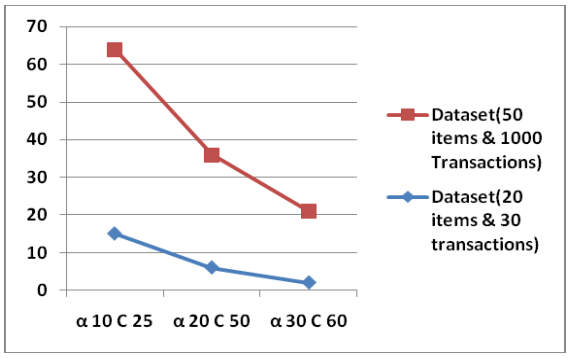

Figure 7. Performance of Genetic Algorithm

Iterations are decreased when maximizing the support and confidence threshold value.

5.6. Time Complexity

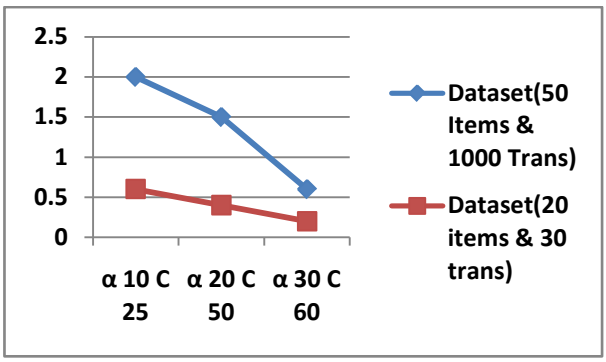

Figure 8. Time Complexity of Genetic Algorithm

The above chart shows the time required for performing modifications of various threshold values in genetic algorithm. Time is decreased when maximizing the threshold values because number of iterations is minimized when maximizing the threshold value.

6. CONCLUSION

Privacy preserving data mining is a new body of research focusing on the implications originating from the application of data mining algorithms to large public databases. In this research work, we have investigated how sensitive rules should be protected from malicious data miner and we 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 this technique, all the sensitive rules are hidden; no false rules can be generated and non sensitive rules are not affected.

7. REFERENCES

[1] Abedelaziz Mohaisen and Dowon Hong. 2008. Privacy Preserving Association Rule Mining Revisited. Journal of the Computing Research Repository.

[2] Atallah, M., Elmagarmid, A., Ibrahim, M., Bertino, E., Verykios. V. 1999. Disclosure limitation of sensitive rules. Workshop on Knowledge and Data Engineering Exchange.

[3] Bikramjit Saikia, Debkumar Bhowmik. Study of Association Rule Mining And different hiding Techniques. Department of computer Science Engineering, National Institute of Technology,Rourkela.

[4] Charu C. Aggarwal and Philip S. Privacy-preserving data mining: Models and Algorithms. ISBN: 0-387-70991-8.

[5] Colin R.Reeves, Jonathan E.Rowe. 2002. Genetic algorithms principles and perspectives.

[6] Darrell Whitley. 1994. A genetic algorithm tutorial. Colorado State University.

[7] Juggapong Natwichai, Xingzhi Sun, and Xue . 2008. A Heuristic Data Reduction Approach for Associative Classification Rule Hiding. Pacific rim international conference on artificial inteligence-PRICAI.

[8] Mohammad Naderi Dehkordi. 2009. A Novel Method for Privacy Preserving in Association Rule Mining Based on Genetic Algorithms. Journal of software-JSW, volume 4,no 6.

[9] Nan Zhang, Shengquan Wang, and Wei Zhao. 2004. A New Scheme on Privacy Preserving Association Rule Mining. Principles of Data Mining and Knowledge Discovery – PKDD, Volume 3202, Pg: 484-495,

[10] Rakesh Agrawal,Tomasz lmielinski,Arun Swami. Mining Association Rules between sets of items in Large Databases.IBM Almaden Research Center,San Jose,CA 95120.

[11] R.R.Rajalaxmi. A Novel Sanitization Approach for Privacy Preserving Utility Itemset Mining", Computer Science and Engineering Kongu Engineering College Erode, TamilNadu, India.

[12] Shyue-Liang Wang Yu-Huei Lee Billis, S. Jafari. 2006. A. Hiding Sensitive items in Privacy Preserving Association rule Mining.

[13] Vaidya, j.Clifton, .W; Zhu, Y.M 2006, X, 121 p, 20 illus, Hardcover..Privacy Preserving Data Mining. ISBN: 979-0-387-25886-7.

[14] Vassilios S. Verykios · Emmanuel D. Pontikakis · Yannis Theodoridis · Liwu Chang. 2007. Efficient algorithms for distortion and blocking techniques in association rule hiding, springer.

[15] Yucel Saygin, Vassilios S.Verkios, Ahmed K. Elmagarmid,. 2002.Privacy Preserving Association Rule Mining. Conference of Research Issues in Data Engineering – RIDE.