Improved Cuckoo Optimization Algorithm for Association Rule Hiding

G. Bhavani, S. Sivakumari

Abstract: ***Association Rule Mining (ARM) is a standard data mining practice used to determine interactions hidden in huge sets. Association Rule Hiding (ARH) methods are used to preserve the privacy of data in ARM. ARH process modifies the original database without changing any non-sensitive rules and data. In order to hide the sensitive rules, cuckoo search optimization algorithm that was developed for hiding the sensitive association rules (COA4ARH) was proposed for sensitive rule hiding. In COA4ARH, number of transactions that should be modified to hide the sensitive rules is not considered which may leads to more number of iteration. In this paper, two properties are introduced to select less number of transactions to be modified. It makes the COA4ARH algorithm faster, decreases the number of lost rules and is suitable for variety of datasets. In order to increase the rule hiding capability of COA4ARH, new fitness functions are introduced. The new fitness functions reduce the amount of lost rules and avoid generation of ghost rules which are formed as objectives of COA4ARH algorithm. The multiple objectives in COA4ARH are conflicting with each other. This is known as multi-objective optimization problem. The multi-objective optimization deals with set of non-dominated solutions (Pareto front) for the problem having more than one objective. It is solved by using Crowding Distance (CD) which selects the optimal set of solution for association rule hiding. Thus, the proposed Improved COA4ARH- CD (ICOA4ARH-CD) can be suitable for variety of datasets and effectively hides the sensitive rules with fewer side effects.***

Keywords: ***Association rule hiding, Cuckoo optimization algorithm, Crowding Distance, Multi-objective optimization problem, Privacy preserving data mining.***

I. INTRODUCTION

Privacy preservation is the primary issue in data mining. In certain applications, wide range of data mining is carried out. In such situation, data mining becomes critical to ensure that information is not uncovered to the public. The sensitive information should be protected in the database once delivering the data to external people. Association Rule Mining (ARM) plays an important role in privacy damage. ARM uses the inference problem to recognize the non-trustworthy data and reveal the trustworthy data. It is solved by using Association Rule Hiding (ARH) [1] techniques in Privacy Preserving Data Mining (PPDM). By preserving sensitive information, the ARH hide the association rules.

Revised Manuscript Received on August 05, 2019.

G.Bhavani, Research scholar, Department of Computer Science and Engineering, Avinashilingam Institute for Home Science and Higher Education for Women, School of Engineering, Coimbatore, India..

S.Sivakumari, Professor and Head, Department of Computer Science and Engineering, Avinashilingam Institute for Home Science and Higher Education for Women, School of Engineering, Coimbatore, India..

ARH [3] is the process of changing the original database with hiding sensitive association rules and without affecting the non-sensitive rules and other data. The objective of the ARH in PPDM is to hide particular information such that it could not be revealed through any ARM algorithm. A COA4ARH [1] used distortion technique to hide the sensitive rules with fewer side effects. In order to escape from any local optimum, an immigration function was introduced in this COA4ARH. Each cuckoo of COA4ARH algorithm optimally distorted the sensitive items to sanitize the original database. However, in COA4ARH algorithm the number of transaction that should be modified for hiding sensitive rules is not considered which may leads to more number of iterations.

In this paper, a minimum number of transactions for modification are decided by introducing two properties based on Minimum Support Threshold (MST) and Minimum Confidence Threshold (MCT). By deciding the number of transactions for modifications, the number of iteration of COA4ARH and the number of lost rules are reduced. The efficiency of COA4ARH algorithm is further improved by introducing new fitness functions which have the capability to reduce the lost rule and avoid the generation of ghost rules. The multiple-objective optimization problem [2] due to new fitness functions is solved by using Crowding Distance (CD) is used in Pareto-optimal solution which hides sensitive rules effectively by selecting optimal set of solution.

II. LITERATURE SURVEY

A new algorithm [4] was proposed for ARH with less complexity and minimum side effects. This algorithm was based on intersection lattice where two heuristics were formulated to hide the sensitive association rules. One of the heuristics, the item to be altered is identified and focuses the sustaining itemsets in the generating set to reduce lost rules. Another heuristic allots weight to each non-sensitive and sensitive rule in a transaction. But, this algorithm requires development in terms of accuracy.

Heuristic based approach based on decrease support of RHS items of Rule clusters [5] was proposed for hiding sensitive association rules. In this approach, owner hid the sensitive association rule and placed transform rules to the server for outsourcing purpose. This technique selects the transactions and items by using some criteria to hide the sensitive association rule. It doesn't disclose the sensitive information in the transaction database. However, this approach needs improvement in terms of side effects.

A new method called Border Rule based Distortion Algorithm (BRDA) [6] was proposed for association rule hiding. The BRDA removed the items whose support and

confidence values are less than a specific threshold value. According to the border rules information, BRDA was chosen the suitable candidates with the low data distortion degree and less side effects. The rules which are weakly relevant were selected preferentially for modifications. However, this algorithm requires high CPU time to hide sensitive rules.

An evolutionary multi-objective optimization [8] algorithm was introduced for association rule hiding. This was achieved by removing the items which were selected based on formulated side effects on missing ghost rules and insensitive rules along with data loss. This optimization algorithm determined appropriate transactions to minimize the side effects. However, the density of dataset affects the performance of this algorithm.

A fuzzy logic approach [9] was proposed for association rule hiding in big data. The appropriate hiding level of each association rule was specified by using a membership degree in fuzzy logic. Moreover, anonymity techniques were utilized to hide the rules instead of deleting the repeated items in the database. In addition to this, parallelization and scalability were applied to make the fuzzy logic approach appropriate for big data analysis. A rule was considered as sensitive rules if and only if its confidence value is greater than confidence threshold. Otherwise, it was considered as non-sensitive rule. However, if there is any changes occurred in member function of fuzzy approach then it causes some change in height of appropriate generalization.

A Least Lion Optimization algorithm (LLOA) [10] was introduced to preserve the privacy of association rules. LLOA was comprised of rule mining stage and secret key generation stage for sanitization. Initially, mine the association rules from the input database by exploring whale optimization algorithm. After that, the rules were validated with a new fitness function. Then, LLOA was applied to hide the association rules by generating a secret key with the help of two factors are utility factor and privacy factor. However, the convergence speed of LLOA was depends on the stopping criterion.

A genetic algorithm for hiding and a technique for creating dummy items [11] were proposed for association rule hiding. These techniques modified the original database with the consideration of that the deleted association rules cannot affect the non-sensitive rules and data. The genetic algorithm for hiding and technique for creating dummy items were used to hide sensitive association rule and sensitive rules. Moreover, for the modified sensitive rules the first technique created dummy items to hide the sensitive association rules. However, the artifactual error rate of this hiding technique is high.

III. PROPOSED METHODOLOGY

Here, an ARH using using cuckoo optimization algorithm is improved by finding a minimum number of transactions for data distortion. Then, new fitness functions are introduced to improve the efficiency of cuckoo optimization algorithm based association rule hiding in terms of side effects. In order to solve the conflicts between the multiple objective function, a Pareto optimal solution with Crowding Distance (CD) is introduced.

A. Selection of less number of transactions for modification to hide a sensitive rule

Privacy preserving ARM focuses on the sanitization of data to the release of trustworthy confidential information. This process is called as data sanitization. The main intention of data sanitization is to minimize the degree of data distortion. The side effects will be minimized by changing minimum number of transactions. To hide the sensitive rules, initially number of transactions that require modifications are determined. This count on modifications is not fixed since it needs to deal with the collection of databases and sensitive rules. So the above count is decided dynamically. A sensitive rule can be secreted by decreasing its confidence or support below the threshold value of minimum support or confidence using MST and MCT respectively. The following two properties are deduced for sensitive rule hiding with less number of transactions:

Property 1

Consider, a set of all transactions $\sum_{a \cup b}$ which support the sensitive rule $a \rightarrow b$. A less number of transactions which are required to be modified in $\sum_{a \cup b}$ is calculated to reduce the support of the rule less than MSR. It is calculated as,

$$NUM_1 = [Supp(a \cup b) \times MST) \times |D|] + 1 \quad (1)$$

In Eq. (1), NUM_1 is the less number of transactions calculated based on MST, $Supp(a \cup b)$ represents the frequency of $a \cup b$ and D denotes the frequency database.

Property 2

Assume, a set of all transactions $\sum_{a \cup b}$ which support the sensitive rule $a \rightarrow b$. A less number of transactions which are required to be modified in $\sum_{a \cup b}$ is calculated to reduce the confidence of the rule. It is calculated as,

$$NUM_2 = [Supp(a \cup b) - Supp(a) \times MCT) \times |D|] + 1 \quad (2)$$

In Eq. (2), NUM_2 is the minimum number of transactions calculated based on MCT, $Supp(a \cup b)$ denotes the frequency of $a \cup b$, $Supp(a)$ denotes the frequency of a and D is a transaction database.

According to the property 1 and property 2, the number of transactions to be modified to hide the sensitive rule $a \rightarrow b$ is inferred as follows

$$Min\{NUM_1, NUM_2\} = Min\{[(Supp(a \cup b) \times MST) \times |D|] + 1, [Supp(a \cup b) - Supp(a) \times MCT) \times |D|] + 1 \quad (3)$$

By the above equation, the number of transactions to be modified for association rule hiding is obtained which is given as input to the cuckoo optimization algorithm.

B. Improved Cuckoo Optimization Algorithm for Association Rule Hiding with New Multi-Objective Fitness function

Initially preprocess the original database to reduce the time consumption and avoid the production of useless and unrelated solutions for association rule hiding. The pre-processing is carried out in two phases. The original database is processed in the first phase where only acute transactions of the database are selected. In the second phase, only the critical sensitive items which need sanitization are altered. After preprocessing and selection of minimum number of alterations, a Cuckoo Optimization Algorithm for Association Rule Hiding (COA4ARH) process is in progress with modifying the population of cuckoo N_{pop}.

Each cuckoo in the population is analytic of a solution (sanitized database). Each cuckoo is shown with a string of 0s and 1s. The 1 and 0 indicates the presence and absence of an item in transaction

respectively. The first cuckoo in the population is a sequence of critical transactions of original database. The other cuckoos randomly calculate the sensitive items and the other items are same as the first cuckoo. Hence, an initial population N_{pop} is generated. The fitness value for the first cuckoo is not necessary because it is the original database. The fitness value of other cuckoos in the initial population is calculated based on number of hiding failure, number of lost rules, rule hiding distance and rule lost distance. In order to improve efficiency of COA4ARH, few more parameters are considered in the fitness value. The association rule hiding can be formulated as multi-objective optimization problem. The multi-objective function is given as follows:

$$Minimize \ \vec{f} = [f_1, f_2, f_3, f_4] \quad (4)$$

$$f_1 = |HF| \quad (5)$$

In Eq. (5), $|HF|$ denotes the number of hiding failure.

$$f_2 = |LR| \quad (6)$$

In Eq. (6), $|LR|$ denotes the number of lost rules.

$$f_3 = RHD + RLD \quad (7)$$

In Eq. (7), *RHD* is hiding distance and *RLD* is the lost distances of rules.

$$f_3 = No.\ of\ GR/R \quad (8)$$

In Eq. (8), *No. of GR* is the number of ghost rule which is a non-sensitive association rule that is not revealed from the original database but can be extracted from the sanitized database and R is the total number of rules that can be mined with the given *MST* and *MCT*.

$$f_4 = No.\ of\ S/\ Size\ of\ D \quad (9)$$

In Eq. (9), the fitness value f_4 defines the data loss, *No. of S* denotes the number of transactions that are sanitized and *Size of D* denotes the size of the database. Based on the new fitness value, move the cuckoos towards the best cuckoos. This process is continued until a user specified number of iterations. Finally the improved COA4ARH (ICOA4ARH) returns the sanitized database with minimum side effects.

C. Multi-Objective Optimization Problem

The multiple objectives are used in the new fitness function of ICOA4ARH. The multiple objective functions are not only interacting with each other but even possibly conflicting with each other. This is called as multi-objective optimization problem and it is formulated as

$$min \ \vec{f}\hat{}(x) = [f_1(x), f_2(x), f_3(x), f_4(x)] \text{ subject to } x \in \Omega \quad (10)$$

In Eq. (10), the decision space is denoted and the decision vector is denoted as $x \in \Omega$.

Multi-objective optimization problem can be solved by two methods. The first method is by integrating all objective functions into a single function. The key problem in this way is to find a proper weight values for each objective function. The second method is by finding a Pareto optimal set. In the proposed method, this optimization problem is solved by finding Pareto-optimal set using crowding distance. A Pareto-optimal solution is a solution, around which there is no way of improving any objective without degrading at least one other objective. The Pareto set is defined and described as,

A vector $X=(x_1, x_2, \ldots, x_{Nobj})$ is said to dominate another vector $X^*=(x^*_1, x^*_2, \ldots, x^*_{Nobj})$, denoted as $X < X^*$, if $\forall\, o \in 1,2,..Nobj$, x_o, x^*_o and $X \neq X^*$.

A feasible solution $x^* \in \Omega$ is called a Pareto optimal solution, if $\exists\, x$ such that $\vec{f}(x) < \vec{f}(x^*)$. The set of all Pareto optimal solutions is called Pareto Set (PS), denoted as,

$$PS=\{ x^* \in \Omega \mid \exists\, x \in \Omega,\ \vec{f}(x) < \vec{f}(x^*)\} \quad (11)$$

The non-dominated cuckoos are in X_i into external repository *rep*. In each iteration, the non-dominated are evaluated one by one to the solution in *rep*. The newly formed solution is added when it is non-dominated by all member of the *rep* or deserted when it is dominated by any member of the *rep*. After this process of addition and removal of new solutions, the still dominating solutions will be deserted. This process is continued until a maximum number of iterations.

In order to enhance the convergence and to produce a well distributed optimal set, Crowding Distance (CD) mechanism has been combined into ICOA4ARH algorithm. The CD value of a solution is evaluated from the density of solutions bordering that solution. CD is computed by arranging the set of solution in decreasing order of the objective function values. The CD value is the average distance between two bordering solutions. The bordering solutions which have the highest (minimum fitness value) and lowest (maximum fitness) objective function values leads to infinite CD values in order to that they are always selected. The overall CD value is evaluated as the sum of discrete distance values conforming to each objective:

$$d_i = \sum_{0=1}^{Nobj} \frac{fi+1 - fi+1}{f\max - f\min} \quad (12)$$

A solution s_1 is said to be constrained-dominated a solution s_2 if any of the following condition is true:

- Both solutions s_1 and s_2 are infeasible, but solution s_1 has a slighter total constraint violation.
- Solution s_1 is feasible and solution s_2 is not.
- Both solutions s_1 and s_2 are feasible and solution s_1 dominates solutions s_2.

In a comparison between two feasible cuckoos, the dominating cuckoo is considered as a better solution. In case of infeasible cuckoos, the cuckoo with a reduced number of constraint violations is considered. The overall process of ICOA4ARH-Crowding Distance (ICOA4ARH-CD) is given as follows.

ICOA4ARH-CD Algorithm

Input: Original Dataset D, R_s, MCT, MST, N_{pop}, $max_{itr\ count}$

Output: Sanitized Dataset D'

1. Select a minimum number of transactions for data distortion using equation (3)

2. Pre-process the original dataset
3. Generate an initial population
4. Each cuckoo in population randomly quantify the sensitive items
5. Calculate the fitness value of each cuckoo using equation s (4) to (9)
6. Find the best solution based on the fitness value
7. repeat
8. Generate new solutions based on *K* and *MR*
9. Limit number of solution to N_{max}
10. for each solution in population
11. Migrate all solutions towards the best solution
12. end for
13. Calculate the fitness value for new best solution using equations (4) to (9)
14. Find the best solution based on the fitness value
15. until the termination condition is satisfied
16. Initialize $itr_{count}=0$
17. Accumulate the non-dominated vectors found in global best solution X_i into *rep*
18. Repeat
19. Compute the CD values of each non-dominated solution in the archive *rep* using equation (12)
20. for $i=1$ to N_{pop}
20a. Randomly choose the global best guide for X_i from a particular top portion of the arranged archive *rep* and save its position into the global best solution.
20b. Calculate the immigration of X_i
20c. $X_i = round(X_i)$
20d. If X_i goes outside boundaries, then it is re-integrated by having the decision variable take the value of its corresponding upper or lower boundary and its velocity is multiplied by -1 so that it immigrates in the opposite direction.
20e. if $(itr_{count} < (max_{itrcount} \times PMUT))$
20f. Perform mutation on X_i
20g. Evaluate X_i
21. End if
22. End for
23. Include all new non-dominated solution in X_i with *rep* when they are non-dominated by any accumulated solutions. All dominated new solutions in the archive are deleted.
24. Determine the solution to be replaced if and only if the archive is full.
24a. Calculate the CD values of each non-dominated solutions in the archive *rep*
24b. Arrange the non-dominated solutions in *rep* in descending CD values
24c. Randomly choose a cuckoo from a specified lowest portion. It consists of the greatest crowded cuckoos in the archive then substitutes it with the new solution.
25. Update the finest solution of each cuckoo in X_i. If the present best solution dominates the position in memory, the cuckoo's position is updated using
26. Global best solution = X_i
27. itr_{count} ++
28. Until $max_{itrcount}$ is reached

In the above algorithm, R_s denotes a set of sensitive association rules, *K* denotes the number of new solutions, *MR* denotes the modification radius, N_{max} denotes the maximum number of persisted solutions with enhanced fitness values, itr_{count} denotes the iteration counter, $max_{itrcount}$ denotes the maximum number of iterations and *PMUT* is the probability of mutation. Based on the above ICOA4ARH-CD algorithm, the input database is sanitized to hide the sensitive rule.

IV. RESULTS AND DISCUSSION

In this section, the efficiency of existing COA4ARH and ICOA4ARH-CD algorithm are tested in terms of hiding failure and lost rule. For the experimental purpose, three databases are chess, mushroom and bank marketing database from UCI machine learning repository were used. The characteristics of databases are given in Table I.

Table I. Database Characteristics

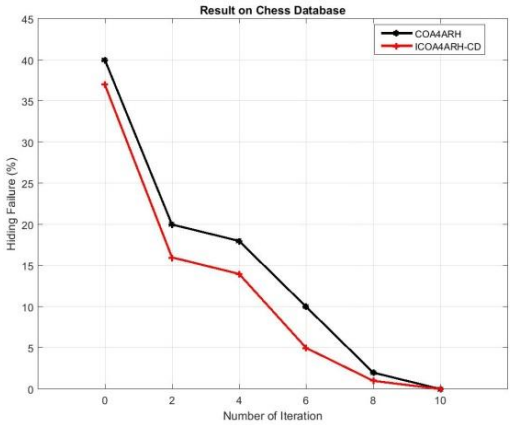
Database	Database Type	No. of Transactions	Avg. Transaction Length	No. of Items
Chess	Real Data	8124	23	119
Mushroom	Real Data	3196	37	75
Bank Marketing	Real Data	4522	17	17

A. Hiding Failure

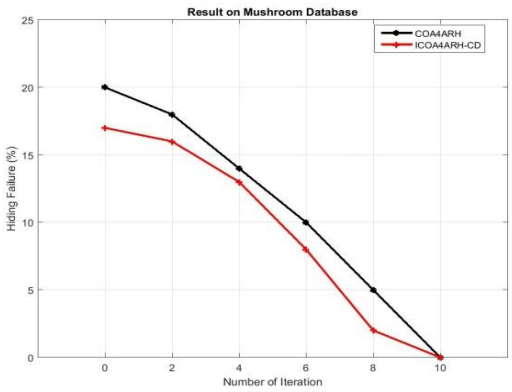
Hiding Failure (HF) denotes the number of sensitive rules which sanitization algorithm could not hide and are still mined from the sanitized data. It is calculated as

$$HF=\frac{|R_s(D')|}{|R_s(D)|} \quad (13)$$

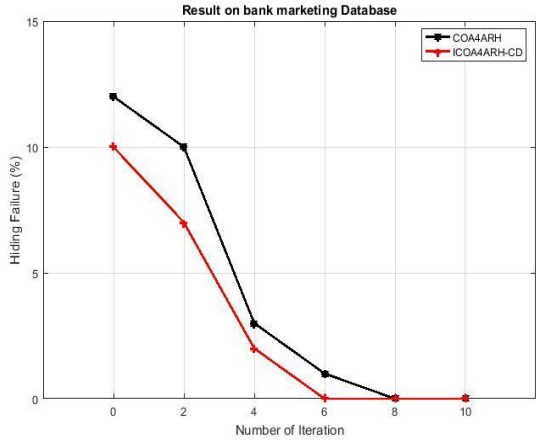
where, the number of sensitive rules discovered in the sanitized database *D'* and original database *D* are denoted as $|R_s(D')|$ and $|R_s(D)|$ respectively.



(a)



(b)



(c)

Fig. 1. Comparison of Hiding Failure. (a) Chess Database, (b) Mushroom Database, (c) Bank marketing Database

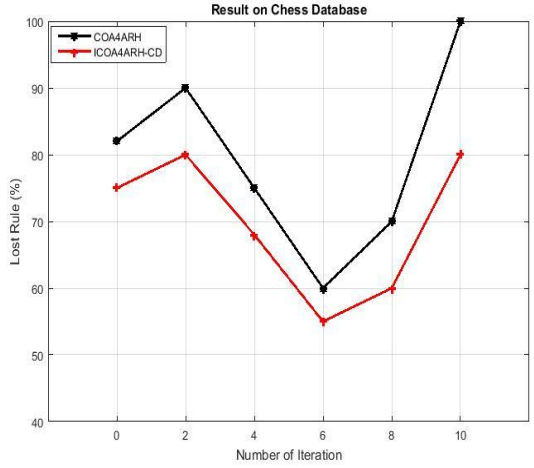
Fig. 1 shows the comparison of hiding failure between existing COA4ARH and proposed ICOA4ARH-CD algorithms on three different databases are chess, mushroom and bank marketing database. When the number of iteration is 4, the hiding failure of ICOA4ARH-CD algorithm is 22.2% less than COA4ARH algorithm in chess database. When the number of iteration of algorithm is 4, the hiding failure of ICOA4ARH-CD algorithm is 7.1% less than COA4ARH algorithm in mushroom database. When the number of iteration of algorithm is 4, the hiding failure of ICOA4ARH-CD algorithm is 33.3% less than COA4ARH algorithm in bank marketing database. From this analysis, it is proved that the proposed ICOA4ARH-CD algorithm has less hiding failure than the COA4ARH algorithm.

B. Lost Rule

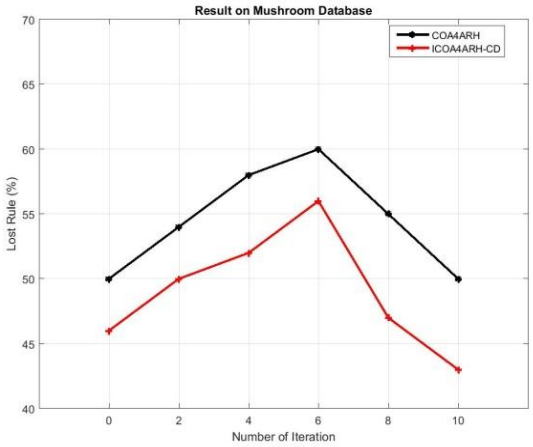
Lost Rule (LR) represents the number of non-sensitive rules that are lost because of the act of sanitization and will not mined from the sanitized database *D'*. It is calculated as,

$$LR=\frac{|\sim R_s(D)|-|\sim R_s(D')}{|\sim R_s(D)|} \qquad (14)$$

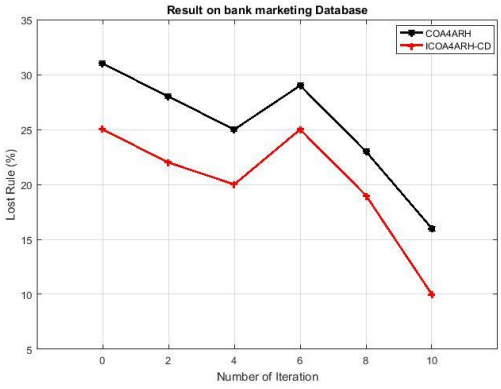
where, the number of non-sensitive rules explored in the original database *D* and sanitized database *D'* is represented as $|\sim R_s(D)|$is and $|\sim R_s(D')|$ respectively.



(a)



(b)



(c)

Fig. 2. Comparison of Lost Rule. (a) Chess Database, (b) Mushroom Database, (c) Bank marketing Database

Fig. 2 shows the comparison of lost rule between existing COA4ARH and proposed ICOA4ARH-CD algorithms on three different databases are chess, mushroom and bank marketing database. When the number of iteration is 6, the lost rule of ICOA4ARH-CD algorithm is 8.3% less than COA4ARH algorithm in chess database. When the number of iteration of algorithm is 6, the lost rule of ICOA4ARH-CD algorithm is 8.3% less than COA4ARH algorithm in mushroom database. When the number of iteration of algorithm is 6, the lost rule of ICOA4ARH-CD algorithm is 13.8% less than COA4ARH algorithm in bank marketing database. From this analysis, it is proved that the proposed ICOA4ARH-CD algorithm has less lost rule than the COA4ARH algorithm.

V. CONCLUSION

In this paper, an Improved Cuckoo Optimization Algorithm for Association Rule Hiding-Crowding Distance (ICOA4ARH-CD) is proposed for efficient association rule hiding. Initially, a minimum number of transactions needs to be modified for data distortion are determined by deducing two properties. Then a new fitness function is introduced in COA4ARH to reduce the amount of lost rules and preserve the algorithms capability of hiding sensitive rules and evading creation of ghost rules. A multi-objective optimization problem is raised due to including more number of objectives in the fitness function. It is solved by introducing Pareto-optimal solution. The convergence of optimal set, Crowding

Distance is combined with ICOA4ARH. Thus the proposed ICOA4ARH-CD method hides the sensitive association rule effectively without the conflicts between multiple objectives and with fewer side effects. The experimental results show that the proposed ICOA4ARH-CD method has better performance in terms of hiding failure and lost rule than the COA4ARH method.

REFERENCES

1. M. H. Afshari, M. N. Dehkordi, M. Akbari. "Association rule hiding using cuckoo optimization algorithm", Expert Systems with Applications, 2016
2. F. Sheikholeslami, N.J. Navimipour, "Service allocation in the cloud environments using multi- objective particle swarm optimization algorithm based on crowding distance", Swarm and Evolutionary Computation, 2017.
3. R. Natarajan, R. Sugumar, M. Mahendran and K. Anbazhagan, K. "Design and Implement an Association Rule hiding Algorithm for Privacy Preserving Data Mining", Int. J. Adv. Res. Comput. Commun. Eng., Vol. 1, No. 7, 2012.
4. H. Quoc Le, S. Arch-int and N. Arch-int, "Association rule hiding based on intersection lattice", Math. Probl. Eng., 2013.
5. P.R. Ponde and S.M. Jagade, "Privacy Preserving by Hiding Association Rule Mining from Transaction Database", IOSR J. Comput Eng. (IOSR-JCE), Vol. 16, No. 5, pp. 25-31, 2014.
6. P. Cheng, I. Lee, J.S. Pan, C.W. Lin and J.F. Roddick, "Hide association rules with fewer side effects", IEICE trans. Inf. Syst., Vol. 98, No. 10, pp. 1788-1798, 2015.
7. S. Mogtaba and E. Kambal, "Association Rule Hiding for Privacy Preserving Data Mining", Ind. Conf. Data Min. Springer, Cham, pp. 320-333, 2016.
8. P. Cheng, I. Lee, C.W. Lin and J.S. Pan, "Association rule hiding based on evolutionary multi-objective optimization", Intell. Data Anal., Vol. 20, No. 3, pp. 495-514, 2016.
9. G.A. Afzali and S. Mohammadi, "Privacy preserving big data mining: association rule hiding using fuzzy logic approach", IET Inf. Secur., Vol. 12, No. 1, pp. 15-24, 2017.
10. D. Menaga and S. Revathi, "Least lion optimisation algorithm (LLOA) based secret key generation for privacy preserving association rule hiding", IET Inf. Secur., Vol. 12, No. 4, pp. 332-340, 2018.
11. S.V. Mohan and T. Angamuthu, "Association Rule Hiding in Privacy Preserving Data Mining", Intern. J. Inf. Secur. Priv. (IJISP), Vol. 12, No. 3, pp. 141-163, 2018.

AUTHORS PROFILE

Ms.G.Bhavani pursuing PhD in Computer Science and Engineering published 2 papers in international journals, her research work include data mining, completed her under graduation at Avinashilingam Institute for Home Science and Higher Education for Women in 2014 and post-graduation at Avinashilingam Institute for Home Science and Higher Education for Women in 2016.

Dr. S. Sivakumari, has received her M.E (Applied Electronics) from Bharathiar University, Coimbatore and PhD (Computer Science and Engineering) from Avinashilingam University for Women, Coimbatore. She is working as Professor in the Department of Computer Science and Engineering, Faculty of Engineering, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, Tamilnadu, India. Her areas of interest are Data Mining, Swarm Intelligence, Neural Networks, Pattern Recognition, Machine Learning. She has published more than 55 technical papers in National, International conferences and International Journals.