Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

Association rule hiding using enhanced elephant herding optimization algorithm

M. Rajasekaran, M.S. Thanabal & A. Meenakshi

To cite this article: M. Rajasekaran, M.S. Thanabal & A. Meenakshi (2024) Association rule hiding using enhanced elephant herding optimization algorithm, Automatika, 65:1, 98-107, DOI: 10.1080/00051144.2023.2277998

To link to this article: https://doi.org/10.1080/00051144.2023.2277998

Published online: 29 Nov 2023.

AUTOMATIKA
2024, VOL. 65, NO. 1, 98–107
https://doi.org/10.1080/00051144.2023.2277998

Association rule hiding using enhanced elephant herding optimization algorithm

M. Rajasekaran[a], M.S. Thanabal[b] and A. Meenakshi[a]

[a]Department of Computer Science & Engineering, Kamaraj College of Engineering and Technology, Virudhunagar, India; [b]Department of Computer Science & Engineering, PSNA College of Engineering and Technology, Dindigul, India

ABSTRACT

Association rule hiding is an efficient solution that helps organizations to avoid the risk caused by sensitive knowledge leakage when sharing data in their collaborations. Cuckoo Optimization Algorithm (COA) sanitizes the transaction database but this method has limitation due to its slow convergence and exploitation capabilities. Hence in this paper, Enhanced Elephant Herding Optimization Algorithm for Association Rule Hiding (EEHOA4ARH) is proposed for association rule hiding. In EEHOA, two core functions such as clan updating operator and separating operator are used for association rule hiding that also realizes the fast convergence and exploitation capabilities. Moreover, the searching strategy in COA4ARH for the selection of best solution is highly time consuming. To reduce the time consumption for the selection of best solution, a Crowding Distance (CD) concept is combined with EEHOA4ARH. By continuously updating the best elephant and replacing the worst elephant in the population, EEHOA4ARH-CD sanitizes the transaction database effectively. Thus the proposed EEHOA4ARH achieves the less computation time, fast convergence and better exploitation capabilities by using crowding distance. The experimental results prove the effectiveness of the proposed EEHOA4ARH–CD method in terms of hiding failure, lost rule and execution time with 44.66 s.

ARTICLE HISTORY
Received 7 August 2023
Accepted 5 October 2023

KEYWORDS
Association rule hiding; evolutionary algorithm; cuckoo optimization algorithm; enhanced elephant optimization algorithm; crowding distance

1. Introduction

Data mining is the task of examining huge volume of data to identify the patterns and sensitive information in it. The data mining process is applied to huge volume of data in companies and business organizations to enable them for making appropriate decisions. Most of the companies and organizations have some sensitive information that should be secured against unauthorized access. Maintaining the privacy or confidentiality of this information is an essential goal for the research area of database security and government organizations. As a result, a key challenge is to find a trade-off between the user's requirements and privacy of information. The use of data mining techniques such as clustering, classification and association rule mining may endanger the database owner's security. Hence, a new research topic called Privacy Preserving Data Mining (PPDM) [1] was introduced.

Nowadays, PPDM has become an important issue since huge volume of personal information has been used by many companies and organizations. In many situations, users are unwilling to reveal their personal data without guaranteeing the protection of their confidential data. To prevent the disclosure of sensitive information, the algorithms in this research area make some modifications in the database (data modifications) and alter amount of data (data distortion) [2] in the database. However, data modification and data distortion protect the confidentiality of sensitive information is not much perfect and has some side effects. Association Rule Hiding (ARH) [3] is a subfield of PPDM that analyses the side effects of data mining methods created from the sensitive information belong to individuals or organizations. The main intention of ARH is to find a sanitized database such that when a mining technique is applied on it, all sensitive rules will be hidden while all non-sensitive rules can be mined.

ARH sanitizes the original database in a way that at least one of the following goals is accomplished [4].

- All the non-sensitive rules that appear when mining the original database at pre-defined threshold of support and confidence can be successfully mined from the sanitized database at the same threshold or higher.
- No rule that is considered as sensitive from the owner's perspective and can be mined from the original database at pre-specified support and confidence, can be revealed from the sanitized database, when this database is mined at the same or at higher thresholds and

CONTACT M. Rajasekaran rajasekaranaravinth@gmail.com Department of Computer Science & Engineering, Kamaraj College of Engineering and Technology, Virudhunagar, Tamil Nadu, India

- No rule that was not derived from the original database when the database was mined at pre-specified thresholds of confidence and support can be derived from its sanitized counterpart when it is mined at the same or at higher thresholds.

One of the powerful and fast algorithms that sanitize a database to hide the sensitive association rules is heuristic approach. Because of their scalability and reliability, most of the researchers in the data mining filed concentrated on heuristic approach [5–9]. But, this approach suffers from unintended side effects and leads them to classify the approximate hidden solutions in various circumstances. So, evolutionary algorithms are used for ARH. Cuckoo Optimization Algorithm for the sensitive Association Rule Hiding (COA4ARH) [10] is an evolutionary algorithm that was used to hide the sensitive association rules with fewer side effects.

Initially, Apriori algorithm was applied on the original database to create the association rules. Then the sensitive rules were selected based on the user defined Minimum Support Threshold (MST) and Minimum Confidence Threshold (MCT). Then the database is pre-processed by selecting the transaction which supported one or more sensitive rules and these are called critical transactions. The sensitive items with critical role in sanitization are addressed for change. Initially, each cuckoo randomly was inserted or deleted the sensitive items in the database to sanitize it. Then new solution was generated based on a fitness function of each cuckoo. Finally, a sanitized database was obtained. However, the whole process of COA4ARH is time consuming, because of the searching strategy used to select the best solution for ARH.

So in this article, an Enhanced Elephant Herding Optimization Algorithm for Association Rule Hiding (EEHOA4ARH) is proposed in which EEHOA is used instead of COA for ARH with less time consumption. EEHOA is also evolutionary-based algorithms which mimics herding behaviour and can be modelled into two operators are clan operators and separating operators. Initially in EEHOA, each elephant in the population randomly insert or delete the sensitive items to sanitize the database. Then fitness functions of each elephant are calculated and the best solution is selected based on clan updating operator and separating operator. Also, the time consumption for the selection of best solution is reduced by introducing crowding distance where an optimal solution is generated by solving the conflicts between the multiple objective functions in fitness function. Thus the proposed EEHOA4ARH-CD reduces the time consumption for ARH and handles the conflict between the multiple objective functions using CD.

The rest of this article is organized as follows: Section 2 studies the research related to ARH. Section 3 explains the proposed EEHOA4ARH to hide the sensitive rules in the database. Section 4 demonstrates the performance efficiency of EEHOA4ARH method. Section 5 summarizes the article with future scope.

2. Literature survey

A rule hiding approach [11] was proposed based on Evolutionary Multi-Objective Optimization (EMO) algorithm for association rule hiding. While sanitizing the database, a balanced relation among the side effects were analysed and collaborated with the association rule hiding process using EMO algorithm. In EMO algorithm, the transactions were determined to be changed through decoding the chromosomes and found the items to be removed. However, the density of the dataset greatly influences the efficiency of rule hiding approach.

A sanitization approach [12] was proposed for hiding sensitive itemsets in the association rule based on the concept of Particle Swarm Optimization (PSO). However, PSO is easy to fall into local optimum in high-dimensional space and has a low convergence rate in the iterative process. A distortion-based method [13] was proposed to hide the sensitive association rules by removing some items in a database which reduced the support or confidence of sensitive rule below user defined threshold values. However, the effectiveness of this method depends on the user defined threshold value.

Modified Decrease Support of LHS item using Equivalent class transformation (MDSLE) [14] approach was proposed for association rule hiding. In MDSLE, the frequent itemsets and the sensitive items in the transactions were identified by applying equivalent class transformation. Also, the sensitive association rules were hiding using a heuristic approach. This approach will be enhanced by decreasing its undesired side effects for achieving less information loss.

An algorithm [15] was proposed to hide the sensitive association rules by inserting dummy items in the rules. However, it is not suitable for huge volume of dataset. A fuzzy logic approach [16] was proposed to sanitize the transaction database. This approach used anonymization approach to hide the sensitive association rules. It avoided the undesired side effects by removing frequent item-sets on new entrance data. The sensitive degree of every association rule was calculated using suitable membership functions and anonymization was done with respect to the membership function. However, if there is any change in membership function, then it causes some change in height of appropriate generalization.

An efficient algorithm [17] was proposed to hide association rule using genetic algorithm. In this algorithm, genetic optimization was used to transform the raw database into sanitized database. It introduced

simple genetic encoding for sensitive association rule hiding. The solution encoding and the objective function were defined for association rule hiding based on genetic algorithm. It achieved a better time cost as well as minimum side effects of the non-sensitive rules. This algorithm will be enhanced by hiding a set of rules in one optimization run instead of hiding one in every run.

An optimization algorithm called Electromagnetic Field Optimization Algorithm (EFO4ARH) [18] was proposed for association rule hiding. At first, electromagnetic particles were generated and each particle indicated a solution to a sanitation database. Then the particles were split into positive, negative and neutral fields based on their fitness function. A new solution was generated based on the position and pole of the chosen particles and after that the fitness value of the new solution was compared with the fitness value of the last solution and if it is better, the last solution would be removed and placed the new solution in the list. In future, a new fitness function will be defined to reduce the number of lost rules.

MAXARH algorithm [19] was proposed for finding the sensitive rules and providing the privacy of sensitive rules. However, this algorithm still has side effects during hiding the sensitive association rules. Whale optimization and Least Lion Optimization Algorithm (LLOA) [20] were introduced for privacy-preserving association rule hiding. LLOA sanitized the database through hiding the sensitive items in the association rules using a privacy factor and utility factor in the objective function of LLOA. But the convergence speed of LLOA depends on the stopping criterion.

Hiding technique based on Genetic Algorithm (HGA) and Dummy Items Creation (DIC) techniques [21] was proposed for hiding sensitive association rules. However, this technique has high artefactual error rate. An efficient meta-heuristic chemical reaction optimization-based algorithm, [22] was proposed for association rule hiding through an advanced perturbation approach. This algorithm combined the characteristics of Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Cuckoo Optimization Algorithm (COA) to perturb the transaction database and hide the sensitive association rules. But it does not show the drastic change in the quality of the sanitized transaction database.

A modified genetic algorithm [23] was proposed for association rule hiding. At first, Frequent Pattern-growth (FP-growth) algorithm was applied to generate association rules and then genetic algorithm was applied on it to hide the sensitive association rules. However, it has slow convergence problem. A pattern sanitization approach [24] was proposed for hiding sensitive itemsets for privacy preserved pattern sharing. However, this approach hides the sensitive itemsets under a single minimum support threshold. An optimized support balance model [25] was proposed for association rule hiding. This model modified the increase shift left-based association rule hiding technique in which only the sensitive rules have the confidence value less than the minimum confidence were hidden. However, this model is not much effective when an association rule has more than two sensitive items.

3. Proposed methodology

In this section, the EEHOA4ARH-CD is described in detail for association rule hiding to preserve the sensitive information in the database. Initially, Apriori algorithm is applied on the collected transaction database D to generate rules. Then the database is pre-processed to avoid the generation of unrelated solution for ARH. During pre-processing critical transaction (i.e. a transaction which fully supports one or more sensitive rules) is selected and then considered only those sensitive items with critical role in sanitization are addressed for change. The sensitive items with critical role are inserted or deleted by EEHOA4ARH to hide the sensitive association rules. The conflict between the multiple objective functions in fitness function can be solved by CD. The block diagram of EEHOA4ARH-CD is shown in Figure 1.

3.1. Enhanced elephant herding optimization algorithm based sanitization of transaction database

Elephants are social creatures live in social structures of claves and females. An elephant clan is headed by a matriarch and consisted of number of elephants. Female elephants in the clan prefer to stay with their family members, whereas the male elephants choose to stay somewhere. They will progressively become independent of their families until they leave their families completely. EEHOA is inspired from the herding behaviour of elephant. Following are some of the assumptions which are considered in EEHOA:

- Some clans with fixed numbers of elephants include the elephant population.
- A certain number of male elephants will abandon their family group and in every generation they live alone far from the main elephant group.
- A matriarch guides the elephants in each clan.

The behaviour of elephants can be modelled as clan updating operator and separating operator. Each elephants in the population either insert or delete the sensitive items in the transactions to sanitize the database. In EEHOA, the behaviour of elephants is modelled as clan updating operator and separating operator. The elephants are updated using their current position and matriarch through clan updating operator and

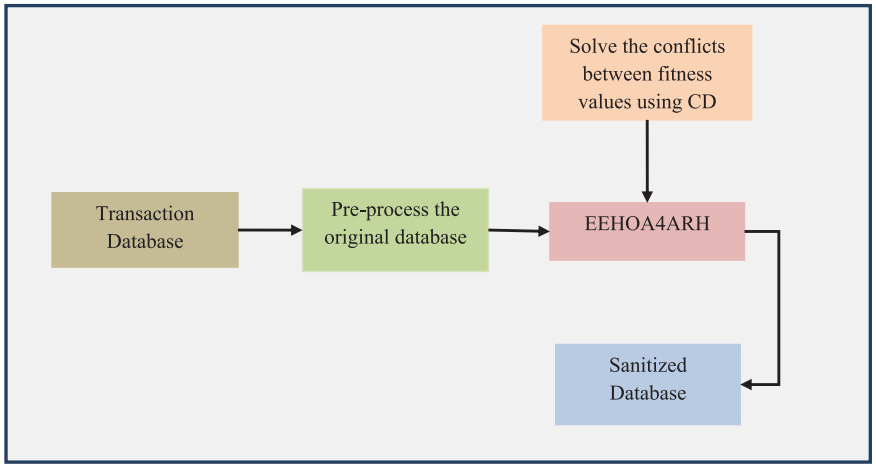

Figure 1. Block diagram of EEHOA4ARH-CD.

the separating operator is then implemented. EEHOA is proposed based on fixing the convergence speed and maintaining a trade-off between exploitation and exploration phases.

a) Initialization and Fitness function

Each elephant in the population in a clan is indicative of a solution (i.e. sanitized database) which is shown with a sequence of 0 s and 1 s. The 1 indicates the presence of sensitive item and 0 indicates the absence of sensitive item in transactions. The first elephant in the population is a sequence of critical transactions of original database. The other elephants in the population randomly quantify the sensitive items and the other items are same as the first elephant. Therefore, an initial population with number of solution is generated. After the initialization process, the fitness values of each elephant are calculated with respect to the number of hiding failure, number of lost rules, rule hiding distance, rule lost distance, ghost rule and data loss. The fitness function is formulated as

$$Min\overrightarrow{fit} = [fit_1, fit_2, fit_3, fit_4, fit_5] \tag{1}$$

$$fit_1 = |HF| \tag{2}$$

$$fit_2 = |LR| \tag{3}$$

$$fit_3 = RHD + RLD \tag{4}$$

$$fit_4 = \frac{No\_of\_GR}{R} \tag{5}$$

$$fit_5 = \frac{No\_of\_S}{Size\_of\_D} \tag{6}$$

In above equations, $|HF|$ denotes the number of hiding failure, $|LR|$ denotes the number of lost rules, RHD is the rules hiding distance, RLD is the rules lost distance, No_of_GR is the number of ghost rule which is a non-sensitive association rule that cannot be discovered from the original database but can be mined from the sanitized database, R is the total number of rules that can be mined with the given Minimum Support Threshold (MST) and Minimum Confidence Threshold (MCT), No_of_S denotes the number of transactions that are sanitized and $Size\_of\_D$ denotes the size of the database.

In each generation, the individual with minimum fitness in a clan c_x is selected as the matriarch (m) at time t.

$$m_x^t = \arg\min_{e \in e_x} \overrightarrow{fit}(e) \tag{7}$$

In Equation (7), e_x is the collection of individual elephants in clan x.

b) Clan updating operator

Every elephant y in clan x has an old position $e_{x,y}^t$. Its new position $e_{x,y}^{t+1}$ is influenced by the clan matriarch m_x^t based on the following equation:

$$e_{x,y}^{t+1} = e_{x,y}^t + \alpha \times (m_x^t - e_{x,y}^t) + \beta \times (c_x^t - e_{x,y}^t) + \gamma \times rand \tag{8}$$

In Equation (8), α, β and γ are scaling factors range from 0 to 1 that finds the influence of the clan matriarch on the elephant new position, affinity of elephant to move towards the clan centre and affinity of elephant to walk randomly, correspondingly. $rand = (2 \times r - 1)(e_{max} - e_{min})$ is a random vector drawn from a uniform distribution, e_{max} and e_{min} are upper and lower bounds of individual elephants position, c_x^t is the centre of the clan and is calculated as

$$c_x^t = \frac{1}{Num_x} \times \sum_y e_{x,y}^t \tag{9}$$

In Equation (9), Num_x is the number of elephants in clan x. To fix the convergence speed, the matriarch update operator in EEHOA is calculated as

$$m^{t+1} = m^t + \beta(c^t - m^t) \tag{10}$$

The matriarch new position is a linear combination of its prior position. Here, the three control factors such

as (α, β, γ) are used to control the convergence towards the clan centre and the random walk in parallel.

c) Separating operator

The separating operator generated by male elephants which can be modelled as

$$e^{t}_{x,worst} = e_{min} + (e_{max} - e_{min}) \times r \qquad (11)$$

In Equation (11), e_{min} and e_{max} are the upper and lower bounds of the elephant individual position respectively and $e^{t}_{x,worst}$ is the worst individual elephant in clan c_x. For the separating operator, probability density function starts with r, a Pseudo Random Number Generator (PRNG) function that creates a uniformly distributed random number in the interval [0, 1]. r has to be scaled and shifted to create a uniformly distributed random number in the range $[e_{min}, e_{max})$. A floor function is used to create a uniformly distributed random integer number in a specified range. It is clear that floor $([e_{min}, e_{max})) = (e_{min}.e_{max-1})$, hence a continuous uniform distribution in the range $[e_{min}, e_{max}]$. The overall flow of EEHOA based sanitization of transaction database is shown in Figure 2.

3.2. Selection of optimal solution based on crowding distance

In COA4ARH, linear searching strategy is applied on the fitness function to select the best solution for sanitizing the database. The linear searching strategy has limitation as high computation time. So in EEHOA, crowding distance concept is used to select the best solution for ARH. The multiple objective functions in fitness function are not interacting with each other and they may be conflicting with each other. This is known as multi-objective optimization problem and it is formulated as

$$\min \vec{fit}(x) = [fit_1(e), fit_2(e), fit_3(e), fit_4(e), fit_5(e)] \quad \text{subject to } e \in \Omega \qquad (12)$$

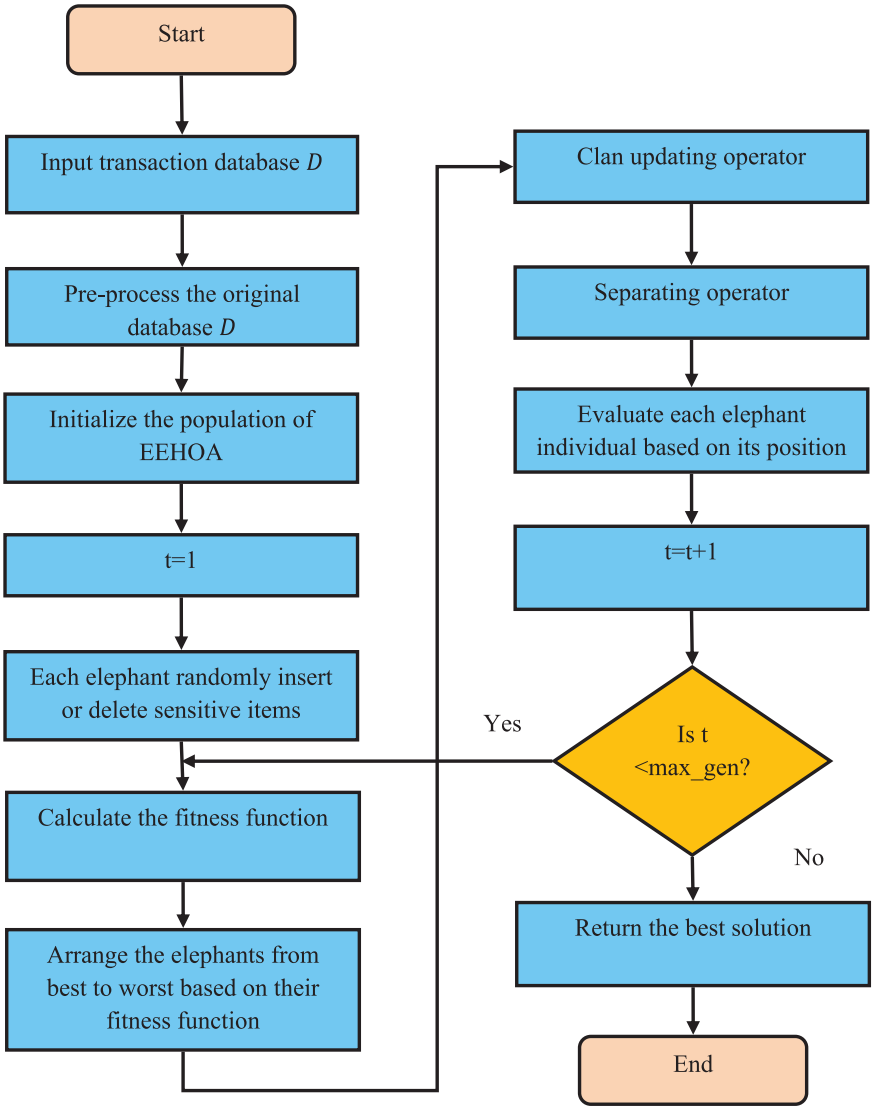

Figure 2. Overall flow of EEHOA-based sanitization of transaction database.

In Equation (12), Ω is the decision space and $e \in \Omega$ is a decision vector.

One of the most popular ways to solve the multi-objective optimization problem is finding a 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 definition and description of Pareto set is given as follows:

A vector $E = (e_1, e_2, \ldots e_{Nobj})$ is said to dominate another vector $E^* = (e_1^*, e_2^*, \ldots e_{Nobj}^*)$, denoted as $E \prec E^*$, if $\forall_\circ \in 1, 2, \ldots Nobj$, $e_\circ, e_\circ^*$ and $E \neq E^*$.

When $\nexists e$ such that $\overrightarrow{fit}(e) \prec \overrightarrow{fit}(e^*)$, a feasible solution $e^* \in \Omega$ is called a Pareto optimal solution. The collection of all Pareto optimal solutions is called Pareto Set (PS), which is given as follows:

$$PS = \{e^* \epsilon \Omega | \nexists se \epsilon \Omega, \overrightarrow{fit}(e) \prec \overrightarrow{fit}(e^*)\} \quad (13)$$

The non-dominated elephants in E_i into external repository *rep*. At every iteration, the non-dominated are compared one by one to the solution in *rep*. When the new solution is dominated by any member of the *rep*, the solution will be leaved. On other hand, the solution will be included to the *rep*. After including the new solution, when there any solutions in the *rep* dominated by the new solution, those solutions will be leaved. This process is continued till a maximum number of iteration is achieved.

To reduce the time consumption and to generate an optimal set, Crowding Distance (CD) mechanism has been combined into EEHOA. CD value of a solution represents an assessment of the density of solutions neighbouring that solution. CD is computed by first arranging the collection of solution in decreasing order of objective function values. CD value of a particular solution is the average distance of its two nearby solutions. The bordering solutions which have the highest (minimum fitness value) and lowest (maximum fitness value) objective function values are given infinite CD values so that they are always chosen. The overall CD value is computed as the sum of individual distance values related to every objective in the fitness function. CD value is calculated as

$$d_i = \sum_{o=1}^{Nobj} \frac{fit_o^{i+1} - fit_o^{i+1}}{\overrightarrow{fit}_o^{max} - \overrightarrow{fit}_o^{min}} \quad (14)$$

A solution s_1 is called as constrained-dominate a solution s_2 when any of the following criterion is true:

- Solution s_1 is sufficient and solution s_2 is not.
- Both solutions s_1 and s_2 are insufficient, but solution s_1 has a smaller overall constraint violation.
- Both solution s_1 and s_2 are sufficient and solution s_1 dominate solution s_2.

When comparing two sufficient elephants, an elephant which dominates the other elephant is considered as a better solution. On other hand, when both elephants are insufficient, the elephant with a less number of constraint violations is considered to be as a better solution.

Enhanced Elephant Herding Optimization Algorithm-Crowding Distance for Association Rule Hiding

Input: Original database D, population size N_{pop}, maximum generation max_*gen*

Output: Sanitized dataset

1. Pre-process the original database D.
2. Generate an initial population of elephant.
3. Each elephant in the population set Rand [0,1] to the sensitive item in the transaction.
4. Calculate the fitness function of each elephant using Equations (1)–(7).
5. while $t \leq$ max_*gen*
6. for $x = 1$ to N_{clans}
7. Arrange clan elephants based on their fitness
8. $e_{x,best}^t =$ first elephant
9. $e_{x,worst}^t =$ last elephant
10. for $y = 1$ to N_x //N_x is the number of elephants in clan x
11. Update each elephant in clan x using Equations (8) and (9).
12. end for
13. Replace the worst elephant in clan x using Equation (11).
14. Evaluate population by the newly updated positions.
15. Increment $t$by 1.
16. end for
17. end while
18. Set $gen_{count} = 0$
19. Save the non-dominated vectors found in E_i into *rep*
20. while $gen_{count} <$ max_gen_{count}
21. Calculate the CD values of each non-dominated solution in the archive *rep* using Equation (14).
22. for $x = 1$ to N_{clans}
23. Randomly choose the global best guide for E_x from a specified top portion of the sorted archive *rep* and store its position to the best elephant.
24. Compute the immigraton of E_x
25. $E_x = round(E_x)$
26. if E_x goes beyond boundaries, then it is reintegrated by having the decision variable take the value of its corresponding lower or upper boundary and its velocity is multiplied by -1 so that it immigrates in the opposite direction.
27. if ($gen_{count} <$ (max_$gen_{count} \times P$)) //P is a probability
28. Perform clan updating operator and separating operator

29. Evaluate E_x
30. End if
31. End for
32. Insert all new non-dominated solution in E_x into *rep* if they are not dominated by any of the stored solutions. All dominated solutions in the archive by the new solution are removed from the archive.
33. If the archive is full, the solution to be replaced is determined
34. Compute the CD values of each non-dominated solutions in the archive *rep*
35. Arrange the non-dominated solutions in *rep* in decreasing order of CD values
36. Randomly choose an elephant from a specified bottom portion which comprise the worst elephant in the archive then replace it with the new solution.
37. Update the best solution of each elephant in E_x. If the current best solution dominates the position in memory, the elephants position is updated
38. $e_{x,best} = E_x$
39. Increment gen_{count} by 1
40. End while

Table 1. Database characteristics.

Database	No. of transactions	Avg. transaction length	No. of items
Chess	8124	23	119
Mushroom	3196	37	75
Bank marketing	4522	17	17

Table 2. Parameter setting of EEHOA4ARH-CD.

Parameters	Values
N_{pop}	80
max_*gen*	100
max_gen_{count}	30
N_x	4
α	0.25
β	0.05
γ	0.015

4. Results and discussion

To analyse the efficiency of EEHOA4ARH-CD, this method is executed on a chess, mushroom and bank marketing database. The proposed EEHOA4ARH-CD is compared with existing COA4ARH and PSO [12] is done with the parameters such as hiding failure, lost rule and execution time. The existing and proposed ARH methods are implemented in MATLAB (Version

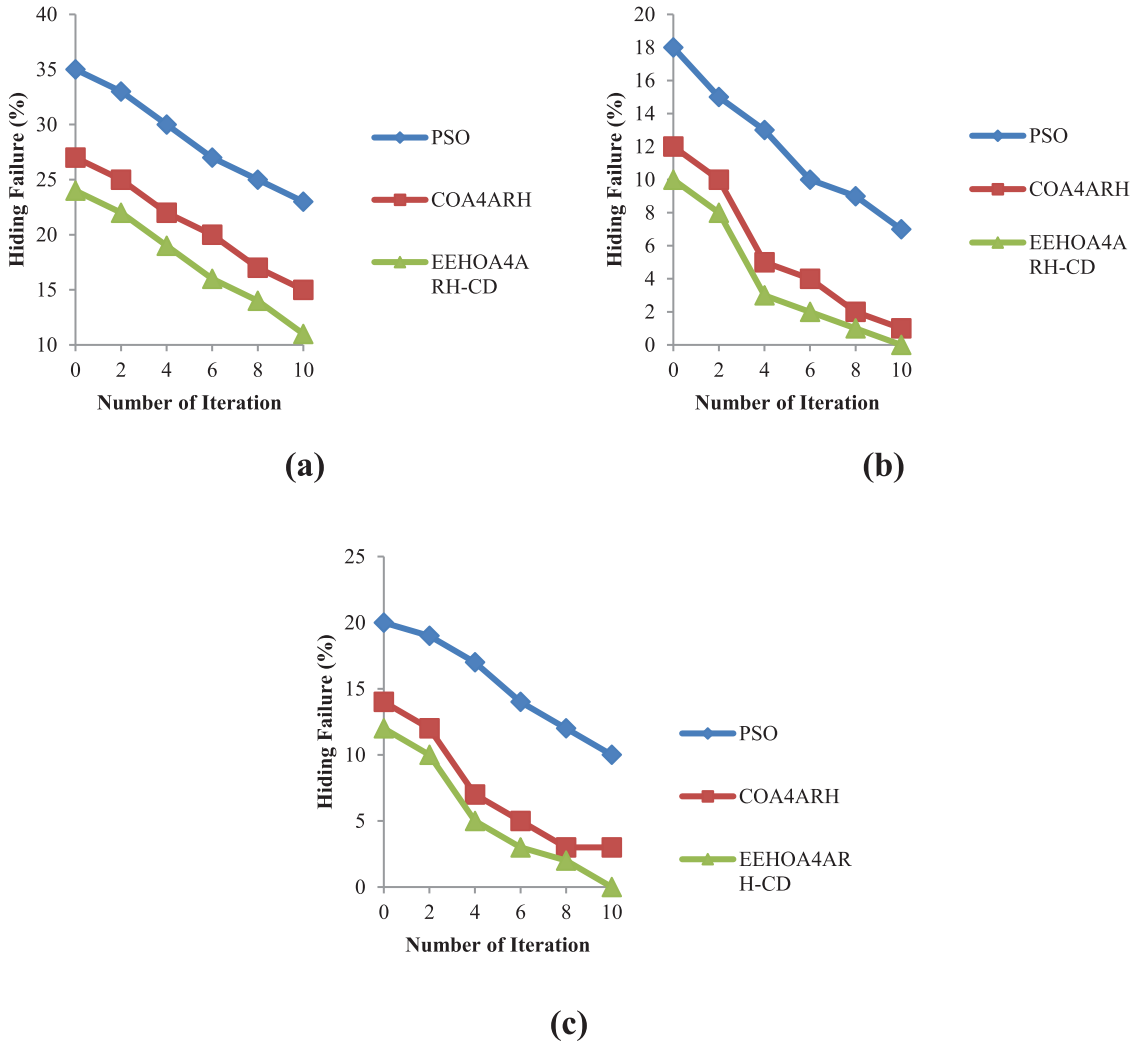

Figure 3. Comparison of hiding failure on (a) chess, (b) mushroom and (c) bank marketing database.

2018a) and runs on a Microsoft Windows 7 with Intel processor running at 2.70 GHz and 4GB memory. The characteristics of chess, mushroom and bank marketing databases is shown in Table 1. Table 2 shows the parameter setting of EEHOA4ARH-CD. The download links of these datasets are provided in [26–28].

4.1. 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. HF is calculated as

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

where $|R_s(D')|$ is the number of sensitive rules explored in the sanitized database D' and $|R_s(D)|$ is the number of sensitive rules explored in the original database D.

Figure 3 shows the hiding failure of PSO, COA4ARH and EEHOA4ARH-CD methods on chess, mushroom and bank marketing database. X denotes the number of iteration and Y axis denotes the hiding failure. When the number of iteration is 6, the hiding failure of EEHOA4ARH-CD is 40.74% and 20% less than PSO and COA4ARH methods on chess database. From this analysis, it is proved that the proposed EEHOA4ARH-CD method has less hiding failure than other methods on three different databases.

4.2. Lost rule

Lost Rule (LR) denotes the number of non-sensitive rules that are lost because of the act of association rule hiding methods. The non-sensitive rule will not mined from the D'. LR is calculated as

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

where $|\sim R_s(D)|$ number of non-sensitive rules explored in D and $|\sim R_s(D')|$ number of non-sensitive rules explored in D'.

Figure 4 shows the lost rule of PSO, COA4ARH and EEHOA4ARH-CD methods on chess, mushroom and bank marketing database. X denotes the number of iteration and Y axis denotes the hiding failure. When the number of iteration is 6, the lost rule of EEHOA4ARH-CD is 37.25% and 11.11% less than PSO

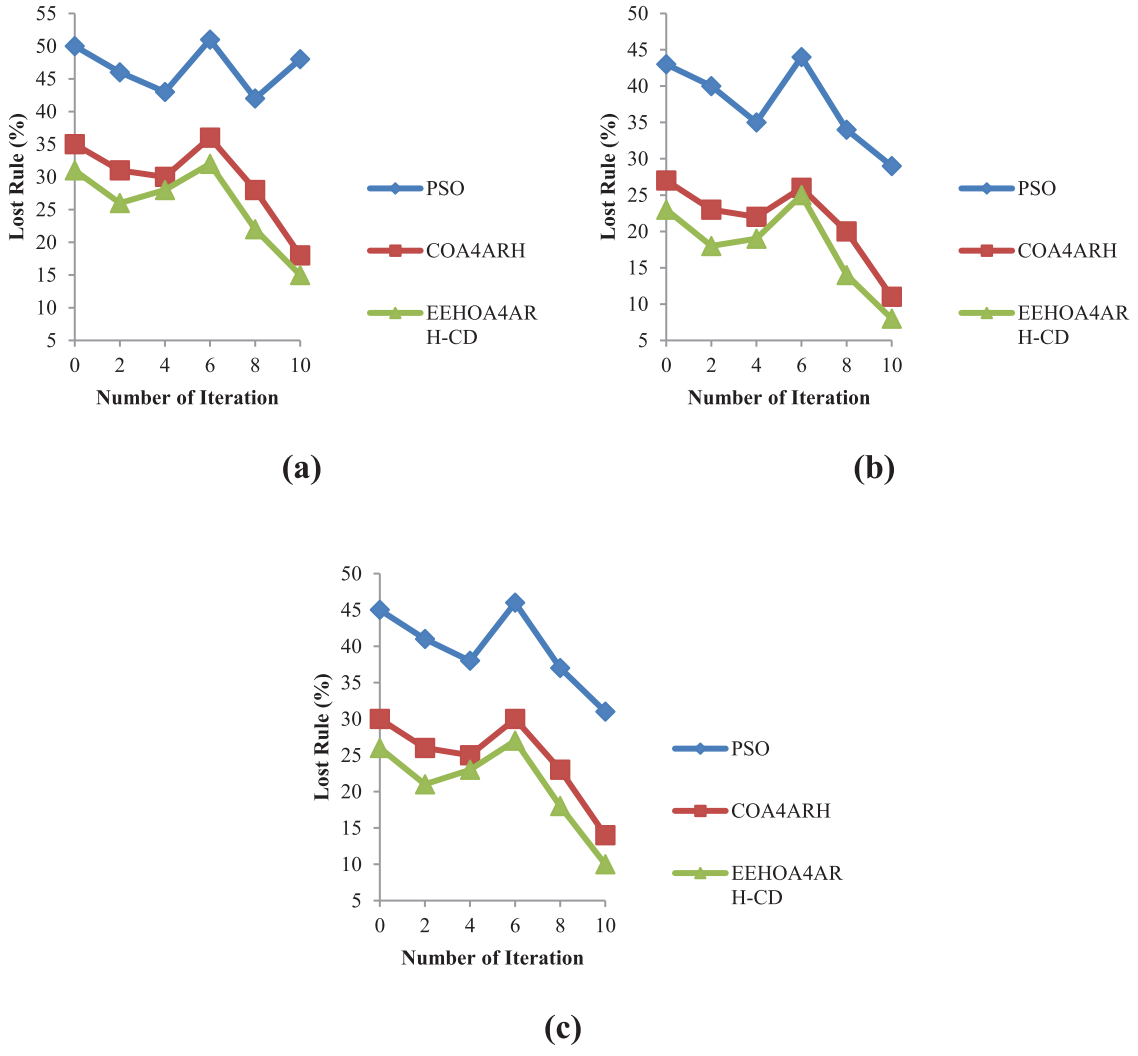

Figure 4. Comparison of lost rule on (a) chess, (b) mushroom and (c) bank marketing database.

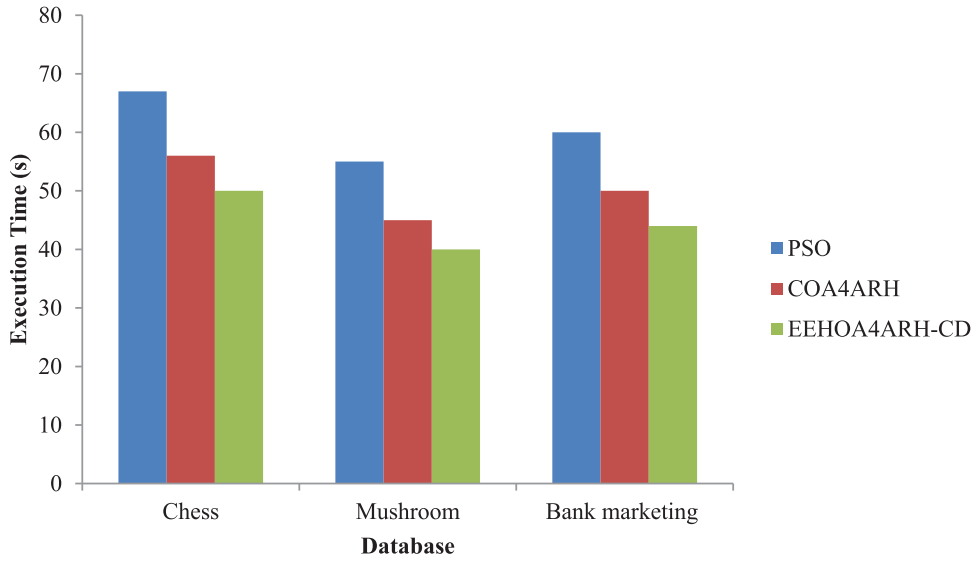

Figure 5. Comparison of execution time.

and COA4ARH methods on chess database. From this analysis, it is proved that the proposed EEHOA4ARH-CD method has less lost rule than other methods on three different databases.

4.3. Execution time

Execution time denotes the amount of time taken by ARH methods to sanitize the transaction database.

The execution time of PSO, COA4ARH and EEHO A4ARH-CD methods on three different datasets is shown in Figure 5. *X* axis denotes the databases and *Y* axis denotes the execution time in seconds. The execution time of EEHOA4ARH-CD method is 25.37% and 10.71% less than PSO and COA4ARH for ARH on chess database. From this analysis, it is proved that the proposed EEHOA4ARH-CD has less execution time than state-of-the-art methods for ARH on three different databases.

5. Conclusion

In this paper, EEHOA4ARH-CD is proposed for ARH with fast convergence rate and exploitation capabilities. In EEHOA4ARH-CD, the clan updating operator has been fixed to avoid the problem of slow convergence and hence improve the exploration phase and this will increase population diversity. The EEHOA4ARH also solved skewed distribution of initial elephant population by using separating operator. The time consumption for selection of best solution is reduced by selecting an Pareto optimal set using crowding distance concept. Finally, the investigational tests on chess, mushroom and bank marketing database proved that the proposed EEHOA4ARH-CD method achieves less hiding failure, lost rule and execution time compared to the PSO and COA4ARH methods. In future, big data analytics can be included.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Mendes R, Vilela JP. Privacy-preserving data mining: methods, metrics, and applications. IEEE Access. 2017;5:10562–10582. doi:10.1109/ACCESS.2017.2706947

[2] Ghalehsefidi NJ. Using distortion and blocking techniques for preventing association rules' discovery. Ind J Sci Technol. 2016;9:18. doi:10.17485/ijst/2016/v9i27/97476

[3] Gopalan NP, Murthy TS. Association rule hiding using chemical reaction optimization. In: Bansal J., Das K., Nagar A, et al., editors. *Soft computing for problem solving*. Singapore: Springer; 2019. p. 249–255.

[4] Sathiyapriya K, Sadasivam GS. A survey on privacy preserving association rule mining. Int J Data Min Knowl Manage Process. 2013;3(2):119–131. doi:10.5121/ijdkp.2013.3208

[5] Ashraf M, Rady S, Abdelkader T, et al. Efficient privacy preserving algorithms for hiding sensitive high utility itemsets. Comput Secur. 2023;132. https://doi.org/10.1016/j.cose.2023.103360

[6] Xu C, Wu H, Liu H, et al. Blockchain-oriented privacy protection of sensitive data in the internet of vehicles. IEEE Trans Intell Veh. 2022;8(2):1057–1067. doi:10.1109/TIV.2022.3164657

[7] Darwish SM, Essa RM, Osman MA, et al. Privacy preserving data mining framework for negative association rules: an application to healthcare informatics. IEEE Access. 2022;10:76268–76280. doi:10.1109/ACCESS.2022.3192447

[8] Suma B, Shobha G. Fractional salp swarm algorithm: an association rule based privacy-preserving strategy for data sanitization. J Inf Sec Appl. 2022;68:103224. doi:10.1016/j.jisa.2022.103224

[9] Menon S, Ghoshal A, Sarkar S. Modifying transactional databases to hide sensitive association rules. Inf Syst Res. 2022;33(1):152–178. doi:10.1287/isre.2021.1033

[10] Afshari MH, Dehkordi MN, Akbari M. Association rule hiding using cuckoo optimization algorithm. Expert

Syst Appl. 2016;64:340–351. doi:10.1016/j.eswa.2016.08.005
[11] Cheng P, Lee I, Lin CW, et al. Association rule hiding based on evolutionary multi-objective optimization. Intell Data Anal. 2016;20(3):495–514. doi:10.3233/IDA-160817
[12] Lin JCW, Liu Q, Fournier-Viger P, et al. A sanitization approach for hiding sensitive itemsets based on particle swarm optimization. Eng Appl Artif Intell. 2016;53:1–18. doi:10.1016/j.engappai.2016.03.007
[13] Cheng P, Roddick JF, Chu SC, et al. Privacy preservation through a greedy, distortion-based rule-hiding method. Appl Intell. 2016;44(2):295–306. doi:10.1007/s10489-015-0671-0
[14] Femandes M, Gomes J. Heuristic approach for association rule hiding using ECLAT. 2017 2nd International conference on communication systems, computing and IT applications (CSCITA); 2017, April, pp. 218–223. IEEE.
[15] Kanekar RP, Dhanaraj R. Adding dummy items to hide sensitive association rules. National Conference On Advances In Computational Biology, Communication, And Data Analytics, IOSR J Comput Eng (IOSR-JCE). 2017:6–10.
[16] Afzali GA, Mohammadi S. Privacy preserving big data mining: association rule hiding using fuzzy logic approach. IET Inf Secur. 2017;12(1):15–24. doi:10.1049/iet-ifs.2015.0545
[17] Khuda Bux N, Lu M, Wang J, et al. Efficient association rules hiding using genetic algorithms. Symmetry. 2018;10(11):576. doi:10.3390/sym10110576
[18] Talebi B, Dehkordi MN. Sensitive association rules hiding using electromagnetic field optimization algorithm. Expert Syst Appl. 2018;114:155–172. doi:10.1016/j.eswa.2018.07.031
[19] Murthy TS, Gopalan NP, Venkateswarlu Y. An efficient method for hiding association rules with additional parameter metrics. Int J Pure Appl Math. 2018;118(7):285–290.
[20] Menaga D, Revathi S. Least lion optimisation algorithm (LLOA) based secret key generation for privacy preserving association rule hiding. IET Inf Secur. 2018;12(4):332–340. doi:10.1049/iet-ifs.2017.0634
[21] Mohan SV, Angamuthu T. Association rule hiding in privacy preserving data mining. Int J Inf Sec Privacy (IJISP). 2018;12(3):141–163. doi:10.4018/IJISP.2018070108
[22] Murthj TS, Gopalan NP, Pudi PP. An efficient meta-heuristic chemical reaction optimization based algorithm for association rule hiding using an advanced perturbation approach. 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS); 2018, June, pp. 1466–1471, IEEE.
[23] Patel J, Shah P. Hiding sensitive association rules using modified genetic algorithm: subtitle as needed (paper subtitle). 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI); 2019, April, pp. 30–34, IEEE.
[24] Surendra H, Mohan HS. Hiding sensitive itemsets without side effects. Appl Intell. 2019;49(4):1213–1227. doi:10.1007/s10489-018-1329-5
[25] Mary G, Reddy AMSS, Agarwal K, et al. Optimized support balance model for association rule hiding. Int J Sci Technol Res. 2020;6(4):2476–2480.
[26] https://www.kaggle.com/datasets/datasnaek/chess.
[27] https://www.kaggle.com/datasets/uciml/mushroom-classification.
[28] https://archive.ics.uci.edu/dataset/222/bank+marketing.