Parallelized BSE-QP-ICOA for ARH

G.Bhavani[a], Dr.S.Sivakumari[b]

[a] Research Scholar, Department of Computer Science and Engineering, Avinashilingam Institute for Home Science and Higher Education for Women, School of Engineering,,Coimbatore

[b]Professor and Head, Department of Computer Science and Engineering, Avinashilingam Institute for Home Science and Higher Education for Women, School of Engineering,,Coimbatore

Abstract: The Association Rule Hiding (ARH) is a traditional method of information shielding which is about changing the real database by removing sensitive rules without altering the quality of it. Balancing Stochastic Exploration-Quality Preserving-Improved Cuckoo Optimization Algorithm for ARH (BSE-QP-ICOA for ARH) was an efficient ARH technique to sanitize the transaction database for ARH. In BSE-QP-ICOA for ARH, a meta-heuristic algorithm called cuckoo search algorithm was used where each cuckoo inserted or deleted the sensitive items based on the multi-objective function for ARH. Because of the enormous volume of data, the implementation of BSE-QP-ICOA for ARH in a single node is no longer useful and it leads to high computational cost issue. It is required to solve these problems by using a distributed architecture. In this paper, MapReduce framework is used to solve the above mentioned problems in BSE-QP-ICOA for ARH. Initially, the transaction database is split into number of independent chunks and it is given as input to map function. In the map function, minimum number transactions for modifications are chosen and its sensitive items are altered depending on the fitness function of each cuckoo. Finally, the map function returns the fitness function of each cuckoo and it is given as input to the reducer function which chooses the best fitness function. According to the finest fitness function, the cuckoo in the map function migrate their position and sanitizes the transaction database. The whole process is named as Parallelized BSE-QP-ICOA for ARH (PBSE-QP-ICOA). Thus, by parallelizing the BSE-QP-ICOA for ARH process using MapReduce the computational cost for ARH is reduced effectively.

Keywords: Association rule hiding, Parallelized association rule hiding, MapReduce, cuckoo search algorithm.

1. Introduction

Recent developments in data mining methods have rendered data from large databases simpler to interpret. At the same time, sensitive information collected from such datasets may subject database owners or their clients to a challenge to privacy. Knowledge hiding avoids malicious data mining **(Shahsavari, et.al.,2014)**. In this hiding major researches related are focused on association rules. Association rules are important methods of determining patterns or irregularities in raw data.

Some rules are identified as sensitive for certain purposes. It should, therefore, be handled in such a way that no adversary will misinterpret such important rules before releasing the database. This can be accomplished by modifying the transaction database so that the important content cannot be mined from the database which is known as Association Rule Hiding (ARH) **(Silvaa,et.al., 2019)**. One of the efficient ARH technique is Cuckoo Optimization Algorithm was for the sensitive Association Rule Hiding (COA for ARH)**(Afshari,et.al., 2016)**. It was a distortion based technique where sensitive items values are modified using cuckoo search algorithm. Nevertheless, the COA for ARH algorithm cannot be used with different databases because that algorithm is subjected to a particular number of transactions for modifications.

In order to fix on a minimum number of transactions and to solve multi-objective optimization problem in COA for ARH, an Improved COA for ARH-Crowding Distance (ICOA for ARH-CD) **(Bhavani, et.al.,2019)** was proposed. The side effects even on non-sensitive rules were reduced by Quality Preserving-ICOA for ARH (QP-ICOA for ARH)**(Bhavani, et.al.,2019)**. It efficiency was further enhanced by using Balancing Stochastic Exploration-QP-ICOA for ARH (BSE-QP-ICOA) **(Bhavani, et.al.,2020)** where variable limits were adjusted dynamically. Moreover, in BSE-QP-ICOA crossover and mutation operators were used in cuckoo search algorithm to maintain the balancing between exploitation and exploration. However, by means of computational cost in terms of storage and time, the BSE-QP-ICOA is high-priced because of its centralized process.

In this paper, the high computation cost problem is concentrated and developing a Parallelized BSE-QP-ICOA (PBSE-QP-ICOA) to access the huge volume data with less computational cost. In order to achieve PBSE-

QP-ICOA, a MapReduce framework is used for ARH. The map and reduce functions are the major functions of the MapReduce framework. Initially, MapReduce splits the data into independent chunks and it is given as input to map function. In the map function, each cuckoos of BSE-QP-ICOA sanitize the transaction database based on their fitness function. The best fitness functions are selected and combined in the reducer function. Thus by using the MapReduce framework in PBSE-QP-ICOA the computation cost for ARH is reduced.Dietary habits are the food choices preferred by persons in their daily life. A healthy dietary habit helps an individual to stay fit and well throughout his life.

2. Literature Survey

Gulwani (2012) proposed an approach for ARH. This approach was focused on data modulation techniques, which altered the positions of the sensitive elements but maintained the same support value. The concept of descriptive rules was used to simplify the rules and then to cover the sensitive rules. A transaction supporting the rule with sensitive item was selected from Representative Rules (RR),. Drop the sensitive item from a chosen transaction and applied the same sensitive item to a partly assisted RR transaction. However, this approach was applicable only on small database.

Khan, A., Qureshi, M. S., & Hussain, A. (2014) suggested an improved genetic algorithm approach for hiding sensitive association rules in a privacy preserving manner. In this approach, new fitness function was constructed based on lost rules and ghost rules. Based on the new fitness function, each chromosome processed crossover and mutation processes to generate new population for ARH. However, the genetic algorithm has slow convergence problem.

Quoc Le, H., Arch-Int, S., & Arch-Int, N. (2013) projected an algorithm for ARH depending upon intersection lattice. In this algorithm, heuristics were devised to point out the affected items depending on the nature of intersection lattice. Also for data sanitization, transactions were found based on their weight values. The derived algorithm covers a specific set of sensitive association rules with low complexity and minimum side effects.

Cheng, P., Lee, I., Pan, J. S., Lin, C. W., & Roddick, J. F. (2015) proposed a border rule based distortion algorithm by deleting certain contents and cover sensitive association rules. It decreased the levels of confidence and support of sensitive rules lower than certain limits. In order to determine the rules that can readily be influenced by database adjustment. Based on positive and negative border rules the secondary transactions were assessed. The loosely connected transactions were chosen for alteration. However, this algorithm has high computation time problem.

Menaga, D., Revathi, S. (2018) introduced Least Lion Optimization Algorithm (LLOA) for ARH. It consisted of two phases such as rule mining and secret key generation. In first phase, whale optimization algorithm was used which validated the association rules with new objective function. In order to provide privacy during association rule mining process, a secret key was generated by using LLOA which is the combination of optimization algorithm and minimum mean square. An optimal secret key sanitized the original database and hide the sensitive association rule. However, the stopping criteria in LLOA greatly influence its convergence speed.

Telikani, A., Gandomi, A. H., Shahbahrami, A., & Dehkordi, M. N. (2020) proposed an Improved Discrete Binary Artificial Bee Colony (IBABC) algorithm for privacy-preserving in ARH. A new region generation mechanism was designed to tradeoff exploitation and investigation in binary ABC algorithm. Then, the IBABC was coupled with ABC to select sensitive transactions for modifications. It modified sensitive transactions instead of deleting the whole transactions. However, IBABC is less scalable for ARH.

3. Proposed Methodology

Here, the future PBSE-QP-ICOA is termed in detail for ARH. A MapReduce framework is used to parallelize the ARH. Initially the MapReduce splits the transaction database into independent chunks. The independent chunks are processed in map and reducer function to sanitize the original database for ARH.

3.1 Association Rule Hiding in a distributed environment

3.1.1 Map function

In MapReduce framework, the transaction database D is processed in the structure of key and value pairs. In the map function, the ID of transaction database is processed as key and the value of item in the transaction database is processed as value. Afterwards, the association rules R_s are created by applying the cuckoo search algorithm to the transaction database. The sensitive association rules are chosen on the basis of the user-specified threshold values. Once the least amount of transactions is chosen on the basis of the properties in [4], the sensitive items including critical role in the transaction are observed. Then, each cuckoo in the population insert or remove the sensitive item in the transactions based on the objective parameters (γ, δ) and sensitivity of transaction.

Based on the values of γ and δ, the sensitive items are removes or inserted in the transaction which even reduce the side effects on non-sensitive rules. In each and every iteration of cuckoo population, the searching space of cuckoos is updated based on the following fitness function

$$\text{Minimize } \vec{f} = [f_1, f_2, f_3, f_4, f_5] \qquad \text{-- (3.1)}$$

In Eq. (3.1), f_1 is the sensitive rule hiding failure; f_2 is the lost rule; $f_3 = \varepsilon + \vartheta$, where ε is the rule hiding distance and ϑ is the rule lost distances, $f_4 = \frac{\mu}{\pi}$, where, μ is the number of ghost rules and π is the total number of rules and $f_5 = \frac{\sigma}{\tau}$, where σ is the number of transactions that are sanitized and τ is the size of the transaction database.

The searching space of each cuckoo is also adjusted by adjusting the variable limits in the modification radius. The conflict between the fitness function is resolved by the Pareto-optimal solution with CD. In order to achieve a tradeoff between the exploitation and exploration, the crossover and mutation operators [6] are used in cuckoo search. At each iteration population size are modified by the linear population reduction process [6]. The whole process is continued in all mappers and it produces an intermediate key and value pair. The intermediate key is the cuckoo ID $(cuckoo\_ID)$ and objective functions ID (fit_ID). The intermediate value is the fitness function $\vec{f}$.

3.1.2 Reducer function

The map function returns the fitness function value of each cuckoo along with its ID and fitness function ID. It is combined in the reducer function of PBSE-QP-ICOA. Also, it selects the best solution and it is resend to the Map function. Based on the best solution, the cuckoo in each mapper migrated to the finest solution. This process is carried to reach maximum iterations. At last, the best solution (i.e., sanitized database) is obtained with less computational cost. The whole process of Map and Reduce function is explained as follows.

```
Function Map $(key: Record_{ID}, value: Record)$

Initialization:

        $Record_{ID} = key$

        $Record = value$

read (cuckoos)

for each cuckoo in the population

        $cuckoo_{ID} = Extract\_cuckooID(cuckoo)$

        $fit = Extract\_f_1, f_2, f_3, f_4, f_5(cuckoo)$

        $\vec{f} = ReturnMinfitness(record, fit)$

$fit_{ID} = m$      //$m^{th}$ cuckoo contains the minimum fitness value

        $new_{key} = (cuckoo_{ID}, fit_{ID})$

        $new_{value} = (min\_\vec{f})$

Emit $(new_{key}, new_{value})$

End for

End Function
```

```
Function Reduce$\left(key: (cuckoo_{ID}, fit_{ID}), value: \left(min_{\vec{f}}, itr < maxitr\right)\right)$
Initialize
        $itr = 1$
for each value in $value$
        $min\_\vec{f} = ExtractMinfitness(value)$
for $(i \coloneqq 1\ to\ pop\_size)$
    for $(j \coloneqq 1\ to\ pop_{size}\ \&\&\ i \neq j)$
      if $\vec{f}(cuckoo_{ID\_i}) < \vec{f}(cuckoo_{ID\_j})$
        $min\_\vec{f} = \vec{f}(cuckoo_{ID\_i})$
        else
            $min\_\vec{f} = \vec{f}(cuckoo_{ID\_j})$
        End if
    End for
End for
        $Finalmin\_\vec{f} = min\_\vec{f}$
 Emit $(cuckoo_{ID}, Finalmin\_\vec{f})$
End function
```

The map and reducer function generates the minimum fitness value and based on it each cuckoo insert or delete the sensitive items in the transaction database. Thus the PBSE-QP-ICOA uses the map and reducer function to sanitize the transaction database with less computational cost.

4. Result and Discussion

Here, the efficiency of BSE-QP-ICOA4ARH and PBSE-QP-ICOA4ARH are analyzed in execution time and memory. For the experimental purpose, three datasets such as adult, bank marketing and hardware store sales dataset are used. Table 1 shows the dataset description.

Table 1. Dataset Description

Dataset	Number of transactions	Number of items	Average transaction length
Adult dataset	32,561	14	15
Bank Marketingdataset	4522	17	17
Hardware store sale dataset	1,00,000	1000	100

4.1 Execution Time

It represents the amount of time taken by ARH algorithms to sanitize the transaction database. The Execution Time (ET) of BSE-QP-ICOA4ARH and PBSE-QP-ICOA4ARH algorithm for different datasets is tabulated in Table 2.

Table 2. Execution Time vs. Dataset

Dataset	Execution Time (s)	
	BSE-QP-ICOA4ARH	PBSE-QP-ICOA4ARH
Adult dataset	50	25
Bank Marketing dataset	20	8
Hardware store sale dataset	25	11

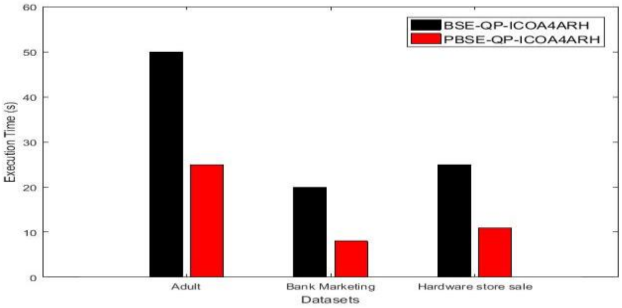

Figure 1. Execution Time vs. Dataset

The ET of BSE-QP-ICOA4ARH and PBSE-QP-ICOA4ARH for three different datasets is shown in Figure 1. The ET of PBSE-QP-ICOA4ARH is 50% less than BSE-QP-ICOA4ARH for adult dataset. From Table 2 and Figure 1, it is proved that the proposed PBSE-QP-ICOA4ARH has less ET than BSE-QP-ICOA4ARH for ARH. Table 3 shows the ET of PBSE-QP-ICOA4ARH algorithm in three different datasets under various numbers of nodes.

Table 3. Execution Time vs. Number of Nodes

Number of nodes	Execution Time (s)		
	Adult dataset	Bank Marketing dataset	Hardware store sale dataset
2	65	30	35
4	45	22	28
6	35	10	15
8	22	8	12
10	10	5	9

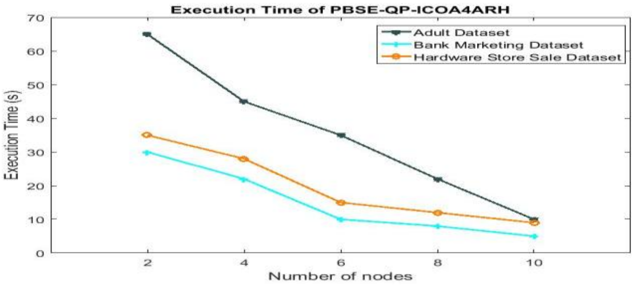

Figure 2. Execution Time vs. Number of Nodes

Figure 2 shows the ET of PBSE-QP-ICOA4ARH under different number of nodes. The number of nodes is taken in X-axis and the ET is taken in Y-axis. From Table 3 and Figure 2, it came to know that by using large number of nodes the execution time of PBSE-QP-ICOA4ARH is reduced.

4.2 Memory Consumption

It denotes amount of memory consumed by BSE-QP-ICOA4ARH and PBSE-QP-ICOA4ARH for ARH. Table 4 shows the memory consumed by BSE-QP-ICOA4ARH and PBSE-QP-ICOA4ARH algorithm for different datasets.

Table 4 Memory Consumption vs. Dataset

Dataset	Memory Consumption (bytes)	
	BSE-QP-ICOA4ARH	PBSE-QP-ICOA4ARH
Adult dataset	300	150
Bank Marketing dataset	100	53
Hardware store sale dataset	120	62

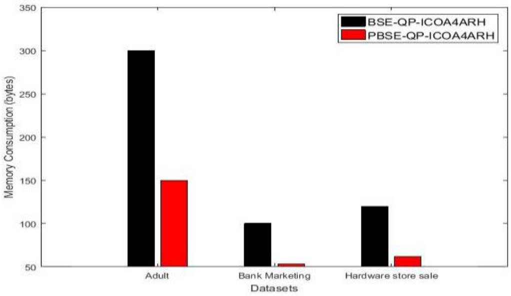

Figure 3 Memory Consumption vs. Dataset

The memory consumption of BSE-QP-ICOA4ARH and PBSE-QP-ICOA4ARH for three different datasets is shown in Figure 3. The memory consumption of PBSE-QP-ICOA4ARH is 50% less than BSE-QP-ICOA4ARH for adult dataset. From Table 4 and Figure 3, it is proved that the proposed PBSE-QP-ICOA4ARH has less memory consumption than BSE-QP-ICOA4ARH for ARH. Table 5 shows the memory consumed by PBSE-QP-ICOA4ARH algorithm in three different datasets under various numbers of nodes.

Table 5 Memory Consumption vs. Number of Nodes

Number of nodes	Memory Consumption (bytes)		
	Adult dataset	Bank Marketing dataset	Hardware store sale dataset
2	227	130	150
4	194	101	120
6	160	63	84
8	127	43	65
10	91	22	35

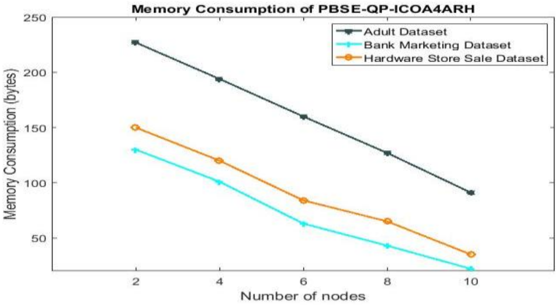

Figure 4 Memory Consumption vs. Number of Nodes

Figure 4 shows the memory consumption of PBSE-QP-ICOA4ARH by using different number of nodes. The number of nodes is taken in X-axis and the memory consumption is taken in Y-axis. From Table 5 and Figure 4, it came to know that by using large number of nodes the memory consumed by PBSE-QP-ICOA4ARH is reduced.

5. Conclusion

In this paper, PBSE-QP-ICOA4ARH is proposed based on MapReduce framework to reduce the computational cost for ARH. The huge volume of transaction database is divided into number of independent blocks and it is processed in map function. In each map function, the minimum number of transactions for distortion is chosen and the sensitive items are inserted or deleted in those transactions based on their fitness function. Each map function returns the fitness function of each block and the best fitness function in the reducer function. Based on the best function, the cuckoos in the map function migrates their position until a maximum number of iterations is achieved. It reduced the computational cost by splitting the whole transaction dataset and processing in different nodes. The experimental results show that the proposed PBSE-QP-ICOA4ARH has less execution time and memory consumption for adult, bank marketing and hardware store sales datasets.

References

Shahsavari, A., & Hosseinzadeh, S. (2014, September). CISA and FISA: efficient algorithms for hiding association rules based on consequent and full item sensitivities. In *7'th International Symposium on Telecommunications (IST'2014)*(pp. 977-982). IEEE.

Silvaa, J., Cubillosb, J., Villac, J. V., Romerod, L., & Solanoe, D. (2019). Preservation of confidential information privacy and association rule hiding for data mining: a bibliometric review. *Procedia Computer Science*, *151*, 1219-1224.

Afshari, M. H., Dehkordi, M. N., & Akbari, M. (2016). Association rule hiding using cuckoo optimization algorithm. *Expert Systems with Applications*, *64*, 340-351.

Bhavani, G., & Sivakumari, S. (2019). Improved cuckoo optimization algorithm for association rule hiding with minimal side effects. *International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8*(10), 337-342.

Bhavani, G., & Sivakumari, S. (2019). Quality Preserving Improved Cuckoo Optimization Algorithm for the sensitive Association Rule Hiding. *International Journal on Emerging Technologies (IJET), 10(4),* 472-477.

Bhavani, G., & Sivakumari, S. (2020). Balancing Stochastic Exploration-Quality Preserving-Improved Cuckoo Optimization Algorithm. *International Journal of Applied Engineering Research(IJAER), 15(4),*425-430.

Gulwani, P. (2012). A Novel Approach for Association Rule Hiding. *International Journal of Advance Innovations, Thoughts & ideas*, *1*(3), 1-9.

Khan, A., Qureshi, M. S., & Hussain, A. (2014). Improved genetic algorithm approach for sensitive association rules hiding. *World Applied Sciences Journal*, *31*(12), 2087-2092.

Quoc Le, H., Arch-Int, S., & Arch-Int, N. (2013). Association rule hiding based on intersection lattice. *Mathematical Problems in Engineering*, *2013*.

Cheng, P., Lee, I., Pan, J. S., Lin, C. W., & Roddick, J. F. (2015). Hide association rules with fewer side effects. *IEICE TRANSACTIONS on Information and Systems*, *98*(10), 1788-1798.

Menaga, D., & Revathi, S. (2018). Least lion optimisation algorithm (LLOA) based secret key generation for privacy preserving association rule hiding. *IET Information Security*, *12*(4), 332-340.

Telikani, A., Gandomi, A. H., Shahbahrami, A., & Dehkordi, M. N. (2020). Privacy-preserving in association rule mining using an improved discrete binary artificial bee colony. *Expert Systems with Applications*, *144*, 113097.