193 *International Journal of Smart Electrical Engineering, Vol.12, No.3, Summer 2023* *ISSN: 2251-9246*
EISSN: 2345-6221

pp. 193:199

Comparing the Speed and Time of Association Rules Extraction from Database with Cuckoo Search and Genetic Algorithms

Payam Abdolmohammadi, Roham Farahani *

[1]Department of Computer Engineering, Islamic Azad University, Arak, Iran, Roham.Farahani@iau.ac.ir

Abstract

This paper aims to study the appropriate data mining method to extract the rules from a data set and examining the benefits of using the cuckoo algorithm to extract association rules and compare the execution time of the cuckoo algorithm and genetic algorithm (GA). Therefore, an algorithm is proposed that includes two parts: preprocessing and mining. The first part presents the procedures related to the calculation of cuckoo fit values and in the second part of the algorithm, which is the main achievement of this research. Support and confidence of the best position can show the most confidence and least support. These mining results can be used to continue mining the association rules. The proposed algorithm is based on the cuckoo search. It hides the sensitive relationship rules with a lower time cost and, at the same time, controls the peripheral effects of non-sensitive rules in a better way. This aim is achieved using recurring to the objective function. The GA is set to be the evaluation criterion to show the prominence of the proposed method. In this method, we compare the speed of the cuckoo algorithm with the genetic algorithm, which uses genetic evolution as a problem-solving model. In general, it is an algorithm based on repetition, most of its parts are selected as random processes, and these algorithms are part of the fitting function. It was chosen as a criterion and we paid. It is scientifically proven that the cuckoo algorithm outperforms the GA in the execution time.

Keywords: Association rules mining, genetic algorithm (GA), cuckoo algorithm, sensitive relationship, non-sensitive relationship, data mining, association rules, data set, time complexity, performance improvement.

Article history: Received 2022-12-06; Revised 2022-11-20; Accepted 2022-12-03, Article Type: Research paper

https://doi.org/10.30495/ijsee.2023.1974141.1240

1. Introduction

Data mining is an effective method for improving decision-making. Many different organizations and institutions apply data mining to extract helpful information and detect latent patterns in their data sets. These applications turned data mining into a promising procedure. Association Rules Mining (ARM) is an important data mining method that extracts common prevalent data from transactional and relational databases to reveal their latent dependencies. The focus of the ARM is to find repetitive items to enable researchers to predict the frequency of an item set depending on the frequency of another item’s frequency in a transaction. The ARM is used in multiple applications including privacy-preserving [1], text document ranking [2], and risk recognition [3] and transaction data sets since 1993 [4].

Our aim in this article is to extract the relationship rules from the transactional data sets and compare the execution time of the association rules in data mining for the cuckoo search algorithm and the GA. Data mining is a technology that uses modelling methods and mathematical algorithms, and machine learning methods to detect unknown and valid patterns in a massive data set. Although this technology is nascent, many companies and organizations including retailers, banks, medical centres, manufacturing factories, telecommunications, and government institutions use data mining tools to analyse their data and detect helpful information and knowledge from them (Rao, 2003). Data mining could extract information from databases that cannot be achieved through queries and logs. The explosive growth of data stored in databases has created the need for new technologies that can intelligently transform huge amounts of data into useful knowledge. Data mining finds the latent patterns in an existing data set semi-automatically (Han, 2005). This technology is different from the other methods of data analysis, in which the system generates patterns from the initial inputs. Data mining extracts important statistical events and structures from massive data sets using algorithmic tools, patterns, variations, anomalies, and rules (Grossman, 1999). It can be said that data mining detects the latent information and relations among current data, and predicts the unknown or unseen relations among them. Mining operation requires data pre-processing. The pre-processing operation includes two parts: information reduction and data summarization and generalization. Reduction means capturing a smaller set from initial data in such a way that mining on the resulting set yields almost the same results as the mining operation on the initial data (Hen, 2007). After performing information reduction and elimination of unrelated features, the summarization and generalization of the data should be performed. Databases often include low-level information; therefore, summarizing such a big data set and representing it as a general concept is crucial. In data generalization, numerous database records are represented conceptually at a higher level. The aim objective of this research is:

- *Main objective:* Comparing the execution time of the association rules in data mining for the cuckoo search algorithm and the GA.
- *Secondary objective:* investigating the execution time of the cuckoo and the GA.

2. Research method

A transaction data set consists of a series of n transactions $DA = (a1, a2, \dots, an)$ .Each transaction is a set of items). $IB = (b1, b2, \dots bm)$ transaction is a pair of an item set and an exclusive transaction identifier: (TID, S). If $Supp\ (S)\ \geq\ Ts$, where Ts is the support threshold defined by the data mining section, item X will be considered repetitive. The $Supp\ (S)$ is the support of the item-sets and indicates the times item-set X occurred in the data set. A rule is presented as $S\ \Rightarrow\ Q$ wherein S and Q are two separate sets from the items. The $S \Rightarrow Q$ rule indicates that S entails another Q in an identical transaction with a certain confidence. Consider the transactions that occurred in a supermarket as an example of a real scenario where the shopping list equals a transaction. If a customer buys “egg” and “cheese”, he or she certainly buys “milk” — the probabilistic relationship rule $Milk \Rightarrow \{egg, cheese\}$. More precisely, $Q \Rightarrow S$ is considered as a relationship rule if $conf\ (S\ U\ Q)\ \geq Tc$ and $Supp(SUQ) \geq Ts$. We define $conf\ (S\ U\ Q)$ as the confidence of S ⇒ Q, that is the support ratio of the rule to the left-hand side of$(S\ U\ Q) = Supp\ (S\ U\ Q)/Supp(S)$. The Ts is the confidence threshold defined by the data mining section. The support and confidence thresholds, $i.e., T_s$ and T_c, are generally related to the transaction group, utilizing the extraction results, and the size of the set. The ARM is the most effective method for extracting the latent patterns and relationships between the items. Personal information – which is often latent in the data set – may be disclosed following performing a reciprocal analysis of the numerous related data sets. Sensitive information disclosure is one of the main side effects of this process. Therefore, privacy has become a vital issue confronting the research society. Today, the association rules are applied on a vast scale, providing numerous algorithms to extract the relationship rules. The cuckoo search algorithm is used in this research. The cuckoo algorithm is capable of performing in both discrete and continuous spaces. It converges very fast (Mareli and Twala, 2018) [20]. On the other hand, the significant role of cuckoo search in optimization problems has ignited this research to use it to mine the association rules using the cuckoo algorithm. In computer science and operations research, a genetic algorithm (GA) is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on biologically inspired operators such as mutation, crossover and selection. In a genetic algorithm, a population of candidate solutions (called individuals, creatures, organisms, or phenotypes) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. This research aims to propose an optimal method using the cuckoo algorithm to improve the association rules factors. The GA is selected to be compared as the criterion. Table 1 shows 24 relationship rules. These rules resulted from a prediction data mining algorithm. If rule *{i}⇒{c,e}* is selected as a sensitive rule, then the victim item would be (e); because it has lower support in comparison to (c): $\alpha(e) = 5$, $\alpha(c) = 6$.

3. Data set description

Three data sets are used in this research for the experiments. Table 2 describes the data sets: Mushroom, Mobile and Chess. Two of the data sets are from the learning repository machine learning. These data sets are originally produced and described in the source. There are six feature variables and one class variable in the chess data set. The second data set Mushroom comprises 8114 records, each of which indicates a single mushroom. The other 22 columns indicate the other different features like the colour, shape, habitat, cloak shape, size, and colour of gills. The third data set Mobile is massive; therefore 5001 random transactions are selected to be studied. This data set included 16 columns, each of which is a feature like location, performance and so on.

4. Genetic algorithms

The concept of GA is aimed to imitate the natural changes that occur in living ecosystems, which is social systems, evaluate the psychological consequences, and model the variable methods. Holland pointed out that Goldberg had significantly contributed to the widespread usage of GA by demonstrating that a large number of problems can basically be solved by utilizing the GA methods. According to, GA is a widely popular search and optimization method for resolving highly intricate problems. The success of its methods has been proven in areas involving machine learning approaches. A complete description of the actual coded GA is provided in this section. Figure 1 provides the flowchart of the standard GA.features such as the implementation of the fitness function are shown. Below is the description of the standard GA procedure: Initial population. This entails the possible solution for set P, i.e., a series of random generations of real values, P = {p1, p2, …, ps}. Evaluation (calculate the fitness value). The fitness function must be delineated in order to evaluate each chromosome in the population, characterized as fitness = g(P). Selection. Following the fitness value calculation, the chromosomes are arranged by their fitness values.

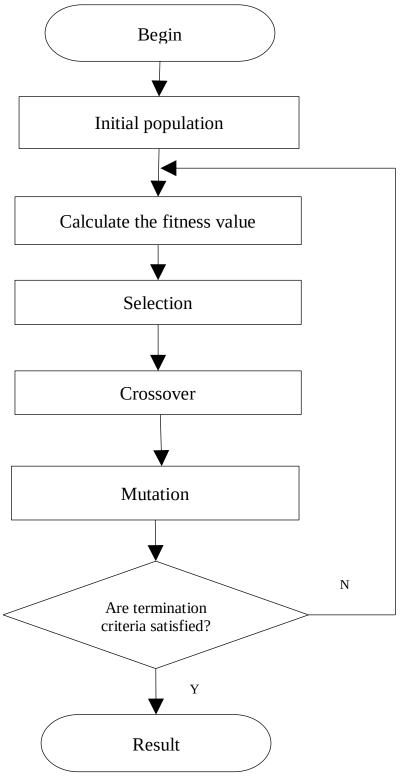

Fig. 1. Flowchart of the standard genetic algorithm (GA)

Table.1.
Relationship Rules - Confidence coefficient - Association rules

Association rules	*Confidence factor*
{i} ⇒ {c}	80%
{i} ⇒ {e}	80%
{e} ⇒ {i}	80%
{e} ⇒ {c}	80%
{k} ⇒ {c}	75%
{r} ⇒ {c}	100%
{u} ⇒ {e}	100%
{f} ⇒ {k}	75%
{k} ⇒ {f}	75%
{f} ⇒ {s}	75%
{s} ⇒ {f}	75%
{k} ⇒ {s}	75%
{s} ⇒ {k}	75%
{i} ⇒ {c, e}	80%
{e} ⇒ {i, c}	80%
{i, c} ⇒ {e}	100%
{i, e} ⇒ {c}	100%
{c, e} ⇒ {i}	100%
{f} ⇒ {k, s}	75%
{k} ⇒ {o, s}	75%
{s} ⇒ {o, k}	75%
{o, k} ⇒ {s}	100%
{o, s} ⇒ {k}	100%
{k, s} ⇒ {f}	100%

Table.2.
Data sets description

Trades Length	*Transactions*	*Items*	*Data set*
37	3169	75	Chess
16	5001	5251	mobile
23	8114	119	Mushroom

The selection of parents is then entailing two parents for the crossover and the mutation. Genetic operators. Once the selection process is complete, the parents' new chromosomes or the offspring (C1, C2) are created by utilizing the genetic operators. The new chromosomes (C1, C2) are then saved into children population C. This process involves the crossover and mutation operations. The crossover operation is applied to exchange information between two parents, which were selected earlier. Several methods of crossover operators are available such as single-point, two-point, k-point crossover, arithmetical crossover... etc. While in the mutation operation, the genes of the crossed offspring's chromosomes are changed. Likewise, several methods are available for the mutation operator.

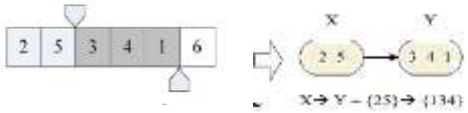

Fig. 2. Chromosomal coding association

5. Proposed algorithm

The proposed algorithm includes two parts: pre-processing and mining. The first part involves the relevant procedures to calculate the cuckoo fit value. Therefore, the data are transmitted and stored in binary format This approach can speed up the database scanning operation and makes backup and reliability selection easier and faster. Then a cuckoo search space is set using the IR value of the item set It uses the cuckoo algorithm in extracting the association rule and calculating the IR value, which is also present in chromosomal coding. The purpose of this action is to generate more meaningful association rules and IR analysis avoids searching for too many association rules that are meaningless item sets in the cuckoo evolution process. This method deals with the points of the front and back parts of each chromosome, and the range decided by these two points is called IR. It is shown in the following equation.

$$\mathrm{IR} = \left[\log\left(\mathrm{mTransNum(m)}\right) + \log\left(\mathrm{nTransNum(n)}\right)\right]\frac{\mathrm{Trans(m,n)}}{\mathrm{TotalTrans}} \quad \mathrm{m} < \mathrm{n} \;\; \text{and}\; \mathrm{m} \neq \mathrm{n} \tag{1}$$

The second part of the algorithm, which is the main contribution of this research, uses the cuckoo algorithm to mine the association rules. First, we start with cuckoo encoding. According to the definition of exploring associative rules, the intersection of the associative rule of the itemset X with itemset Y,x→y Items that appear in itemset X do not appear in itemset Y and vice versa. Hence, front and back points must be given for chromosome coding, the itemset before the partition point is called itemset X, the itemset between the front and back partitions is called the Y itemset. String coding is the chromosomal coding method used in this study. In this example, the highest value of IR,"$IR\{2" \rightarrow "5\}$" It means the lowest point of the front partition, and the most rear partition points are 2 and 5, respectively. This stage is like the chromosome encoding in the GA. In this study, the fit value is used to evaluate the importance of each particle. The fit value of each particle is obtained from the fit function Here, we use the objective function presented by Kang to determine the value of the fit function of the study with the Eq.2.

$$Fitnee(k) = confidence(k) X \log(sipport(k) X \; length(k) + 1) \tag{2}$$

Fit (k) is equal to the fit value of the associative rule of type k. The confidence (k) is equal to the confidence of the associative rule of type k. Support (k) is equal to the true support value of the associative rule of type K. The length (k) is equal to the length of the associative rule of type k. The purpose of this function is to maximize the fit. The greater the support and certainty of the particle, the greater the associative power and it means that it is an important associative rule. Support, reliability and item set length in equal must be calculated before calculating the fit value. The next stage is to generate a population of cuckoos according to the calculated fitness value. Finally, the search procedure continues until the stopping criterion is met which means the optimal habitat is found, i.e., it shows the least support and the most confidence. Therefore, these parameters can be used to extract the association rules and valuable information. Figure 3 shows the structure of this algorithm.

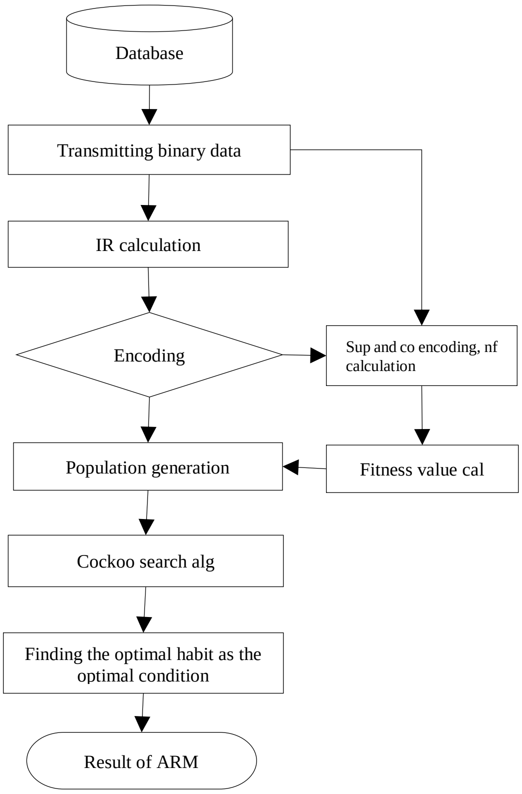

Fig. 3. Proposed algorithm for ARM

6. Proposed model

Table 3 shows different confidence thresholds in the association rule extraction stage.

Table.3.
The experimental adjustments of data sets

Data set	***Sensitive***	***Association***	***Min confidence***
chess	72	7177	94%
	62	6145	95%
	45	4435	96%
	41	4081	97%
mushroom	50	4949	45%
	45	4402	50%
	40	3937	55%
	36	3535	60%
	20	2092	40%
mobile	19	1984	50%
	18	1801	50%
	15	1580	70%
	13	1333	80%
	12	1246	90%

First, the data mining algorithm is applied to these data sets using different values of support and confidence thresholds. Then, 1% of the resulting rules are randomly selected as sensitive rules. Finally, two algorithms were used to hide the sensitive data, and the following results were produced. Multiple experiments are performed for each data set, in which the sensitive rules were always set to 1% and were selected randomly each time. Table 3 shows different confidence thresholds in the association rules extraction stage. The following charts indicate the proposed cuckoo search algorithm outperforms the GA. The execution time of both of the algorithms is drawn against the confidence threshold for the data sets. The execution time is the time an algorithm spends executing a program. This incremental confidence threshold decreases the sensitive rules; it is clear that the proposed algorithm spends less time over all data sets in any conditions compared to the GA.

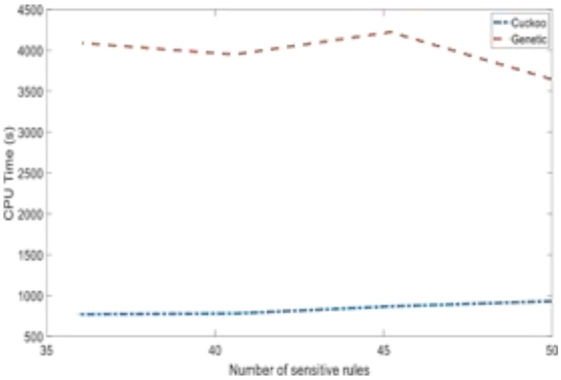

Fig. 4. Mushroom data set

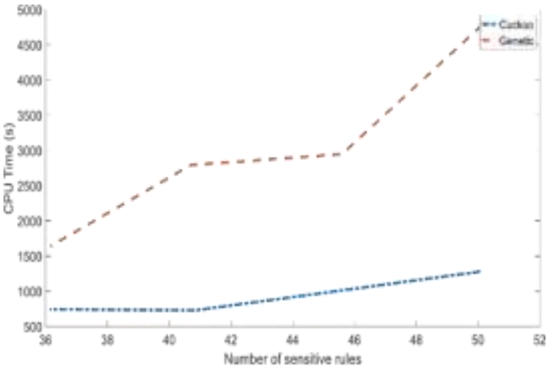

Fig. 5. Chess data set

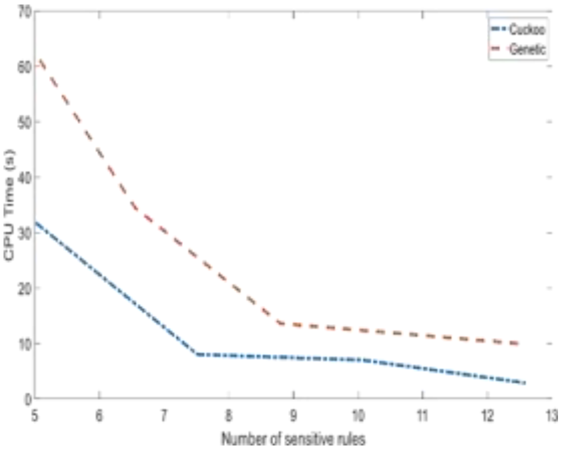

Fig. 6. Mobile data set

The above charts compare the execution time for the Mushroom data set (figure 3), Chess data set (figure 4), and Mobile data set (figure 5). The experiments show that the proposed algorithm is faster than the GA. The performance improvement is shown in table 4.

Table.4.
Performance improvement

Improvment Time	***Max***	***Mobile***	***Chess***	***Mushroom***
	81%	48%	70%	81%

7. Conclusion

In this research, we have developed a new algorithm based on cuckoo-search and a method for improving associative rule factors with the cuckoo algorithm that of the cuckoo to extract the associative rule and we used a module to extract the best fit value. In order to demonstrate the superiority of our approach, Genetic algorithm was chosen as the criterion. The process of this algorithm was very similar to the genetic algorithm but the presented procedures only include coding, calculate the fit value, population production, searching for the best model, and they were a stop condition. Each of these steps in the cuckoo algorithm and the process of generating associative rules was explained that incremental confidence thresholds it leads to a reduction in the number of sensitive rules and it is clear that there is less execution time for the proposed algorithm Compared to the genetic algorithm for all data sets required in all situations. we managed to achieve this by using the cuckoo algorithm and using return in its objective function. As a result, this new algorithm based on cuckoo search was presented to hide sensitive association rules with better time cost. Scientific analysis showed that the performance of the cuckoo algorithm was better compared to the genetic algorithm in terms of execution time.

References

[1] Rehman, S.; Sharma, A. Privacy-Preserving Data Mining Using Association Rule Based on Apriori Algorithm. Int. Adv. Inform. Comput. Res. 2017, 712, 218–226

[2] Srivastava, N.; Gupta, K.; Baliyan, N. Improved Market Basket Analysis with Utility Mining. In Proceedings of the 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT), Jaipur, India, 26–27 March 2018.

[3] Liu, L.; Yu, S.; Wei, X.; Ning, Z. An improved Apriori-based algorithm for friends recommendation in a microblog. Int. J. Comput. Syst. 2018, 31, e3453.

[4] Hareendran, S.A.; Chandra, S.V. Association Rule Mining in Healthcare Analytics. In Data Mining and Big Data; Springer: Cham, Switzerland, 2017; pp. 31–39.

[5] Ning, C.P.; Ji, X.; Wang, H.Q.; Du, X.Y.; Niu, H.T.; Fang, S.B. Association between the sonographer's experience and diagnostic performance of IOTA simple rules. World J. Surg. Oncol. 2018, 16, 179.

[6] Petrova, E.; Pauwels, P.; Svidt, K.; Jensen, R.L. In search of sustainable design patterns: Combining data mining and semantic data modeling on disparate building data. In Advances in Informatics and Computing in Civil and Construction Engineering; Springer: Cham, Switzerland, 2019; pp. 19–26.

[7] Rekik, R.; Kallel, I.; Casillas, J.; Alimi, A.M. Assessing web sites quality: A systematic literature review by text and association rules mining. Int. J. Inf. Manag. 2018, 38, 201–216.

[8] Jabri, S.; Dahbi, A.; Gadi, T.; Bassir, A. Ranking of text documents using TF-IDF weighting and association rules mining. In Proceedings of the 2018 4th International Conference on Optimization and Applications (ICOA), Mohammedia, Morocco, 26–27 April 2018; pp. -6.

[9] Divanis, A.G.; Verykios, V. An integer programming approach for frequent itemset hiding. In Proceedings of the 15th ACM International Conference on Information and Knowledge Management, Arlington, VA, USA, 5–11 November 2006; pp. 5–11.

[10] Agrawal, R.; Imielinski, T.; Swami, A. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD international conference on Management of data, Washington, DC, USA, 25–28 May 1993; pp. 207–216.

[11] Yogendra, J.K.; Vinod, K.Y.; Geetika, S.P. An Efficient Association Rule Hiding Algorithm for Privacy Preserving Data Mining. Int. J. Comput. Sci. Eng. 2011, 3, 2792–2798.

[12] Sui, P.; Li, X. A privacy-preserving approach for multimodal transaction data integrated analysis. Neurocomputing 2017, 253, 56–64.

[13] .Farkas, C.; Jajodia, S. The inference problem: A survey. ACM SIGKDD Explor. Newslett. 2001, 4, 6–11. Available online: http://www.utdallas.edu/~muratk/courses/course_material/p6-farkas.pdf (accessed on 27 June 2018). [CrossRef]

[14] Aqra, I.; Herawan, T.; Ghani, N.A.; Akhunzada, A.; Ali, A.; Razali, R.B.; Choo, K.K.R. A novel association rule mining approach using TID intermediate itemset. PLoS ONE 2018, 13, e0196957.

[15] Mohan, S.V.; Angamuthu, T. Association Rule Hiding in Privacy-Preserving Data Mining. Int. J. Inform. Secur. Priv. 2018, 12, 141–163

[16] Moustakides, G.V.; Vassilios, V.S. A MaxMin Approach for Hiding Frequent Itemsets. In Proceedings of the Sixth IEEE International Conference on Data Mining—Workshops, Hong Kong, China, 18–22 December 2006.

[17] Sun, X.; Philip, S.Y. Hiding Sensitive Frequent Itemsets by a Border-Based Approach. J. Comput. Sci. Eng. 2007, 1, 74–94.

[18] Telikani, A.; Shahbahrami, A. Optimizing association rule hiding using combination of border and heuristic approaches. Appl. Intell. 2017, 47, 544–557.

[19] Le, H.Q.; Arch-Int, S. A Conceptual Framework for Privacy Preserving of Association Rule Mining in E-Commerce. In Proceedings of the 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), Singapore, 18–20 July 2012

[20] Le, H.Q.; Arch-Int, S.; Nguyen, X.Y.; Arch-Int, N. Association Rule Hiding in Risk Management for Retail Supply Chain Collaboration. Comput Ind. 2013, 64, 77.784.

[21] Le, H.Q.; Arch-Int, S.; Arch, N. Association Rule Hiding Based on Distance and Intersectio Lattice. Math. Probl. Eng. 2013, 210405

[22] Lin, C.W.; Zhang, B.; Yang, K.T.; Hong, T.P. Efficiently hiding sensitive itemsets with Sci. World J. 2014, 2014, 398269.

[23] Lin, C.W.; Hong, T.P.; Yang, K.T.; Wang, S.L. The GA-based algorithms for optimizing hiding sensitive itemsets through transaction deletion. Appl. Intell. 2015, 42, 210–230.

[24] Lin, C.W.; Hong, T.P.; Wang, J.P.; Lan, G.C. A GA-Based Approach to Hide Sensitive High Utility Itemsets. Sci. Word J. 2014, 804629.

[25] Hong, T.P.; Lin, C.W.; Chang, C.C.; Wang, S.L. Hiding sensitive item sets by inserting dummy transactions. In Proceedings of the 2011 IEEE International Conference on Granular Computing, Kaohsiung, Taiwan, 8–10 November 2011; pp. 246–249.

[26] Oliveira, S.R.M.; Osmar, R.Z. A Unified Framework for Protecting Sensitive Association Rules in Business Collaboration. Int. J. Bus. Intell. Data Min. 2006, 1, 247

[27] Verykios, V.S.; Elmagarmid, A.K.; Bertino, E. Association Rule Hiding. IEEE Trans. Knowl. Data Eng. 2014, 16, 434–447.

[28] Dhyanendra, J. Hiding Sensitive Association Rules without Altering the Support of Sensitive Item(S). Int. Conf. Comput. Sci. Inf. Technol. 2012, 3, 500–509.

[29] Kalyani, G.; Chandra, S.R. Particle Swarm Intelligence and Impact Factor-Based Privacy Preserving Association Rule Mining for Balancing Data Utility and Knowledge Privacy. Arab. J. Sci. Eng. 2017, 43, 4161–4178.

[30] Prabha, M.; Sathiya, S.V. Association rule hiding using artificial bee colony algorithm. Int. J. Comput. Appl. 2011, 33, 41–47.

[31] Gupta, M.; Joshi, R.C. Privacy-Preserving Fuzzy Association Rules Hiding in Quantitative Data. Int. J. Comput. Theory Eng. 2009, 1, 1793

[32] Ji, C.R.; Deng, Z.H. Mining Frequent Ordered Patterns without Candidate Generation. In Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), Haikou, China, 24–27 August 2007; pp. 402

[33] Zaki, M.K.; Srinivasan, P.; Ogihara, M.; Li, W. New Algorithms for Fast Discovery of Association Rules. In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA, USA, 14–17 August 1997; pp. 283–286. Bayardo, R. Efficiently Mining Long Patterns from Databases. 1998, pp. 85–89. Available online: https://archive.ics.uci.edu/ml/datasets.html