Hiding Sensitive Medical Data Using Simple and Pre-Large Rain Optimization Algorithm through Data Removal for E-Health System

M. Madhavi[1*], Dr.T. Sasirooba[2] and Dr.G. Kranthi Kumar[3]

[1*]Research Scholar, Annamalai University, Chidambaram, Tamil Nadu, India.
madhavi.macharapu@gmail.com, Orcid: https://orcid.org/0000-0002-3540-9813

[2]Assistant Professor, Department of Computer Science and Engineering, Annamalai University, Chidambaram, Tamil Nadu, India. sasiruba@gmail.com,
Orcid: https://orcid.org/0000-0001-9544-4496

[3]Sr. Assistant Professor, Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddartha Engineering College, Vijayawada, A.P, India.
kranthi@vrsiddhartha.ac.in, Orcid: https://orcid.org/0000-0001-8370-8474

Received: March 04, 2023; Accepted: April 08, 2023; Published: May 30, 2023

Abstract

Privacy has become a significant factor of e-Health system in the area of data mining termed as Privacy preserving data mining (PPDM) as it can uncover underlying rules and hide sensitive data for data sanitization. Various algorithms and heuristics have been studied to hide sensitive data using transaction removal. However, they are facing challenges to attain the reasonable side effects. Thus, rain optimization algorithm (ROA) based sensitive data hiding techniques is proposed in this paper. Using this algorithm, suitable transactions to be removed are selected. Besides, in this work, ROA based two frameworks are designed for data sanitization that are simple ROA to remove transaction (sROA2RT) and pre-large ROA to remove transaction (pROA2RT). In this algorithm, fitness is evaluated based on four side effects such as hiding failure, artificial cost, missing cost and dissimilarity of database. The proposed frameworks are evaluated using three e-Health datasets. Compared to previous frameworks, the proposed frameworks attain reasonable side effects.

Keywords: PPDM, ROA, sROA2RT, pROA2RT.

1 Introduction

In general, hospitals store a large amount of patient health data, including family medical history, chronic diseases, medications, dosing, vaccinations, and so on. It is extremely difficult to manage and maintain the massive amount of health data collected from patients.

However, the growing volume of public health data necessitates the development of a secure and collaborative framework that will enhance data transparency and assist the public health ministry in providing the most reliable access. Nevertheless, centralising of those kind large amounts of sensitive data and granting third-party access to such data raises privacy issues.

Journal of Internet Services and Information Security (JISIS), volume: 13, number: 2 (May), pp. 177-192.
DOI: *10.58346/JISIS.2023.I2.011*

*Corresponding author: Research Scholar, Annamalai University. Chidambaram, Tamil Nadu, India.

With the rapid development of data mining schemes in recent decades, significant data can be easily obtained to assist doctors in e-Health system (Aggarwal et al ,2008), (Chen et al ,1996), (Agrawal et al,1993). Various data mining techniques help to obtain the necessary data from many specific requests; Knowledge gained from these techniques can be categorized into association rules, regularization methods, classification, clustering, and application mining. Association-rule mining is a method applied to find relationships between attributes in large databases (Dede et al ,2020), (Yang et al, 2020), (Wenzhe et al, 2017).

Conventional data mining methods evaluate databases to discover possible relationships between attributes. Few applications require protection from disclosure of personal, confidential or secure data such as patient's ID, symptoms, and treatment history. These data contains confidential information and may pose a privacy threat if misused. Several heuristic algorithms (Evfimievski et al ,2002), (Wu et al ,2019), (Lin et al, 2015), (Wu, J. M. T et al, 2021) have been presented to choose the suitable data to filter in the original database to completely hide important data. Along with the personal data the scope of proprietary information can extend to e-Health systems. Some of the information shared between hospitals can be extracted and used by other doctors, which can increase benefits for organizations, but also threatens the spread of sensitive data.

PPDM offers to reduce the privacy threat by hiding sensitive data and allows the extraction of necessary information from the database. When important information is completely hidden, there are serious side effects such as lost costs and artificial costs. The optimal selection of data in the process of data sanitization is to be an NP-hard problem. (Lin et al,2015) first applied the genetic algorithm (GA) for PPDM to sanitize the sensitive data. Several new PPDM algorithms and frameworks have been actively developed recently. Although those models are more useful than standard methods of data cleansing, need to attain reasonable side effects. Thus, this work has the objective to present a heuristic optimization algorithm with less processing time for data sanitization.

- For hiding the sensitive data in the database, rain optimization algorithm (ROA) is presented in this paper. As the algorithm has good convergence speed and solution selection time, it is chosen. This algorithm is used to select the suitable transactions to be removed.
- Besides, based on the ROA, two frameworks such as sROA2RT and pROA2RT are developed.
- In this algorithm, fitness is evaluated in terms of the side effects. Along with three side effects, dissimilarity of the database is also considered as the one of the factors for fitness calculation.

The following sections organize the paper as follows. Section 2 reviews the articles which presented sensitive data hiding techniques. Section 3 proposes Hiding Sensitive Medical Data Using Simple and Pre-Large Rain Optimization Algorithm through Data Removal for E-Health System. Performance of the proposed frameworks is analysed in terms of side effects in section 4. Section 5 concludes the work.

2 Related Works

This section focuses to review the recent literatures which presented different sensitive data hiding techniques. Among them, (Wu, J. M. T et al,2021) presented GA-based system for hiding sensitive medical information. In this approach, varied threshold values were set depend on the varied lengths of sensitive patterns. The authors have chosen a tighter threshold to offer better security of sensitive data as the length of the pattern increases. It leads to protect the identity of the patients in the e-Health datasets. For data sanitization, they designed two model based on GA using record deletion methods. Because of the proposed scheme, the authors achieved better outcomes in terms of side effects.

(Lin, J. C. W et al,2019) proposed a sanitization method to hide sensitive data in the environment of IoT. In the sanitization method, the authors included the hierarchical-cluster scheme. Besides, they presented multi-objective particle swarm optimization algorithm for balancing the side effects such as database dissimilarity, artificial cost, hiding failure and missing cost during the process of data sanitization. They compared their proposed model with the approaches based on multi-objective NSGA-II and single-objective cpGA2DT. By presenting the proposed model, the authors achieved better results in terms of hiding failure (Mileva, A., 2022).

(Lin, J. C. W et al,2019) proposed a multi-objective algorithm termed as Non-dominated Sorting GA to delete transaction abbreviated as NSGA2DT with two models to hide the confidential data by the deletion of transaction using the NSGA-II model. In this approach, the authors considered database dissimilarity as the one of the factors for balancing the side effects. The proposed multi-objective model provided more solutions than single objective algorithms. Thus, the proposed has the flexibility to find the suitable transactions to be deleted based on the preference of users. Besides, the authors presented Fast Sorting strategy (FSR) to select the optimized transaction to be deleted pre-large concept for speed up the execution. Due to the proposed schemes, the authors attained reasonable side effects.

Collaborative data mining can sometimes reveal sensitive patterns within data that may not be desirable to the data owner. Sensitive Pattern Hiding (SPH) is a subset of data mining that solves this issue. Nevertheless, most methods used to hide sensitive models have high side effects on insensitive models, reducing the usefulness of the sanitized dataset. Moreover, most of them are sequential in nature and cannot handle large quantity of data and often result in high processing time. To overcome these non-feasibility and utility issues, (Sharma et al, 2020) presented two parallelized methods known as parallelized grouped victim item removal abbreviated as PGVIR and parallelized hiding candidate removal abbreviated as PHCR. These methods performed depending on spark parallel computing system. These proposed methods caused lesser side-effects to the data.

There are various mechanisms proposed by researchers using evolutionary methods for sensitive data hiding. These evolutionary methods hide sensitive patterns by deleting sensitive transactions. The biggest challenge facing the algorithm is failure of sensitive models and loss of data. The efficiency of the evolutionary algorithm decreases further when applied to dense data sets. Thus, (Jangra et al,2020) presented PSO inspired algorithm based on victim item deletion abbreviated as VIDPSO for sanitizing the dense datasets. Every particle in the algorithm has n count of sub-particles defined from already computed victim items. This algorithm was highly exploratory to find the solution space to choose the best transactions. Because of the proposed scheme, data loss was reduced. In the past, several heuristics-based approaches were introduced to sanitize the sensitive data in PPDM. They mess with the original database to hide sensitive data using delete or additional actions. This is termed as NP-hard problem. Thus, (Ahmed, U et.al, 2021) proposed data privacy in heterogeneous IoT networks. Namely, the authors presented a technique based on deep re-enforcement learning for sanitizing the sensitive data of the dataset. Besides, the authors considered known side effects to be minimized. Experimental analysis of the proposed scheme was executed on both real time and synthetic datasets. Results of the article depicts that the proposed scheme attained reasonable hiding failure results.

3 Proposed Methodologies

3.1. Preliminaries

Consider $L = \{L_1, L_2, ..., L_n\}$ as set of n attributes in a recognizable health dataset D. The dataset D is included with set of records about recognizable information i.e., $D = \{R_1, R_2, ..., R_m\}$. Each attribute

has r_q values as well as one empty value i.e., $L_q = \{\phi, l_1^q, l_2^q, ..., l_{r_q}^q\}$. Besides, the dataset D has set of transactions which is denoted as $D = \{t_1, t_2, ..., t_m\}$, here each transaction has set of records. A minimum support threshold is predefined and denoted as η. Consider the support count of attribute as sup (L_j). A set of attributes can be considered as $freq(L_j)$ if sup (L_j) is greater or equal to η and defined in (1), i.e.,

$$freq(L_j) = \frac{\sup(L_j)}{|D|} \geq \eta \qquad (1)$$

The main role of PPDM is to hide the sensitive attributes with lesser side effects Figure 1

Illustrates the relationship between the attributes before and after the process of PPDM. In the figure, H denotes the large attribute set of D, I denotes the sensitive attributes described by users that are large, I' denotes the non-sensitive attributes that are large and H' denotes the large attributes after the deletion of few transactions.

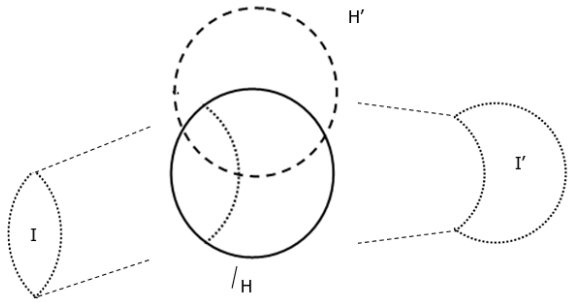

Figure 1: Relationship between the Attributes Before and After the Process of PPDM

Definition 1: The hiding failure is described as the failure to hide the sensitive attributes after the process of data sanitization. It can be denoted as α. Here, $\alpha = I \cap H'$. The value of α should be zero. Figure 2 illustrates the side effect of α.

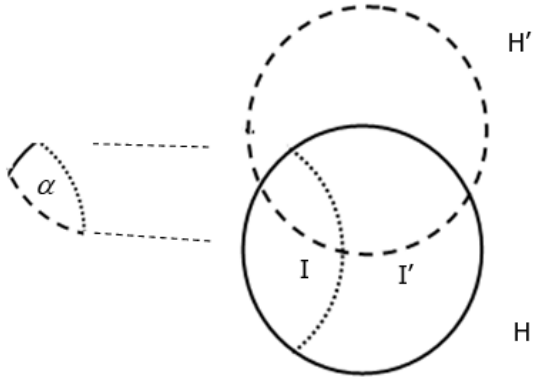

Figure 2: Side Effect of α

Definition 2: The missing cost is described as the attribute which is not retrieved from the database. This missing attribute is a non-sensitive large attribute in the actual database. The missing cost is denoted as β. Here, $\beta = I' - H' = (H - I) - H'$. Figure 3 illustrates the side effect of β.

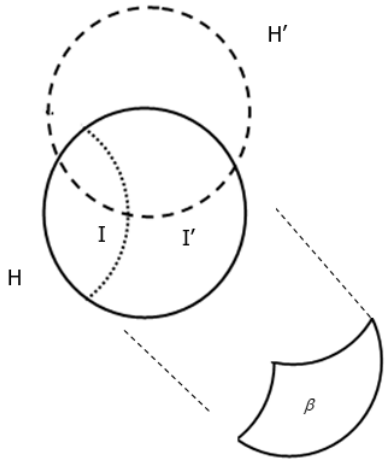

Figure 3: Side Effect of β

Definition 3: The artificial cost is defined as the set of large attributes in the sanitized dataset may not appear in the large attribute of original dataset. It can be denoted as γ . Here, $\gamma = H' - H$. Figure 4 illustrates the side effect of γ .

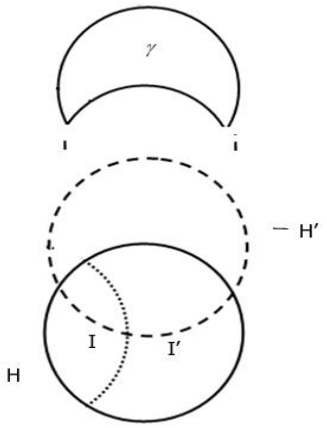

Figure 4: Side Effect of γ

Definition 3: Along with these side effects, another measurement is considered that is known as dissimilarity of database. It measures the count of removed transactions between the original and sanitized database. It can be defined in (2) as follows,

$$Dis = |D - D'| \quad (2)$$

Here, D and D' denote the original and sanitized database respectively.

3.2. Data Hiding Using ROA Algorithm

To select the suitable transactions to be removed for hiding sensitive attributes, ROA based model of sROA2RT and pROA2RT is presented in this paper. Consider set of sensitive attributes to be hidden as defined in (3).

$$\phi = \{sl_1, sl_2, ..., sl_k\} \quad (3)$$

To hide the sensitive attributes using the proposed algorithm via transaction removal the support count of a sensitive attribute should be less than the threshold of minimum support. Here, each transaction to be removed should have any of the sensitive attributes in ϕ . Namely, D' is attained from D, where each t_j in D' should have any of the sensitive attributes in ϕ . The selection of suitable transactions to be removed using ROA algorithm is described as follows:

The ROA is a nature-inspired algorithm that is designed based on the behavior of raindrops. This algorithm is useful for searching and finding an optimal solution from a large search domain within an acceptable CPU time. The steps involved in the data hiding process are explained below;

Step 1: Solution initialization: In this technique, all raindrops in a population indicate the solution. Initially, we generate the solution randomly. The initialization of j^{th} drop is described in (4) below,

$$D_{rop}^{j} = \left\{S_1, S_2, , S_3 \ldots . S_j\right\} j \in [1,2,3,\ldots\ldots,N] \quad (4)$$

Where, the size of the population is denoted as N, S_j denotes the candidate solution or raindrop. In this work, the candidate solution represents the suitable transactions to be removed for hiding sensitive attributes. The candidate solution S_j is represented in (5);

$$S_j = \{t_i\}_j, \quad t_i \subseteq D^{'}; 1 \le l \le z \quad (5)$$

Here, t_i denotes the transaction to be removed. z denotes the count of removed transactions and it can be calculated by summing the difference between the count of sensitive attributes and the minimum support count as in (6).

$$z = \sum_{i=1}^{k} \left(freq(sl_i) - (\eta * |D|) + 1 \right) \quad (6)$$

Step 3: Fitness calculation: We evaluate each solution's fitness after the solution initialization process is finished. Fitness of each solution is defined depend on hiding failure, artificial cost, missing cost and dissimilarity and is defined in (7)

$$Fit_j = \min(\alpha * \omega_1 + \beta * \omega_2 + \gamma * \omega_3 + Dis * \omega_4) \quad (7)$$

Here, $\alpha, \beta, \gamma \, and \, Dis$ denote the hiding failure, missing cost, artificial cost and dissimilarity respectively. $\omega_1, \omega_2, \omega_3 \, and \, \omega_4$ denote the weighting parameters.

Step 4: Update the solution: For updation processes following steps are utilized. During optimization, a random point can be generated near the drop. j^{th} drop's neighborhood point q is described as NP_q^j . A neighbor point of the drop is created based on the following condition (8).

$$\left\| \hat{u}_G * \left(G^j - NP_n^j \right) \right\| \le \left\| \hat{u}_m * r \right\| \quad for \quad \begin{matrix} j = 1,2,3,\ldots., s \\ n = 1,2,3,\ldots., np \\ m = 1,2,3,\ldots., p \end{matrix} \quad (8)$$

Where np represents the number of neighbouring points and $\hat{u}_p$ represents the unit vector of the p^{th} dimension, r represents the magnitude of the neighbourhood. The r is calculated using equation (9).

$$r = f(itr) * r_{initial} \quad (9)$$

Where, *f(itr)* represents the function utilized to limit the size of the neighbourhood within iterations and $r_{initial}$ represents the early dimensions of the neighbourhood.

Dominant drop: The leading neighbour point (NP_d^j) is one of the points selected from the drop's (G^j) neighbour point, which satisfies the below condition (10) and (11),

$$Fit(G^j) > Fit(NP_d^j) \quad (10)$$

$$Fit(NP_q^i) > Fit(NP_d^i) \quad (11)$$

Process of explosion: The explosion procedure is started to address the drop's position if it does not have enough nearby points or is unable to continue the search process to reach the optimal minimum. Using equation (12), we can generate the number of neighbouring points in this explosion process.

$$np_E = be \times np \times ce \quad (12)$$

Where, the number of neighbouring points without the explosion process is represented as np the explosion counter is represented as ce and the number of neighbouring points generated in the explosion process is represented as np_E

The rank of raindrop: Every iteration uses (13), (14) and (15) to determine rank (R) for every raindrop. The merit order list uses this rank.

$$V1_t^j = Fit(G^j)\big|_{at\ t^{th}\ iteration} - Fit(G^j)\big|_{at\ first\ iteration} \quad (13)$$

$$V2_t^j = Fit(G^j)\big|_{at\ t^{th}\ iteration} \quad (14)$$

$$Ra_t^j = order(V1_t^j) * \tau_1 + order(V2_t^j) * \tau_2 \quad (15)$$

Where, the difference between the fitness function of drop D^j at the first iteration and t^{th} iteration is represented as $V1_t^j$, the fitness function of drop D^j at t^{th} iteration is defined as $V2_t^j$, $order(V1_t^j)$ and the orders of V1 and V2 at t^{th} iteration is denoted as $order(V2_t^j)$ when they are organized in ascending order, the weighting coefficients are denoted as τ_1 and τ_2 , that are assumed as 0.5 and the rank of a rain drop is described as R_t^j .

List of merit order: For each iteration, the ranks of the raindrops are organized in order of ascending. A drop with a low ranking may be taken from the list, and some drops may be granted important rights. The optimal solution or map function is the drop with the minimum fitness function.

Step 5: Termination: The above process is repeated till the minimum fitness function is achieved. Otherwise, the algorithm is terminated. The ARO algorithm is presented in table 3

3.3. Algorithm for sROA2RT and pROA2RT

The sROA2RT algorithm includes a simple ROA algorithm for hiding sensitive attributes by transaction removal. Initially, high frequent attributes H is attained in line 1. From line 2 to 6, projection of transactions having any of the sensitive attributes is described. From 7 to 10, initialization of solutions is described. Here, solution represents the transaction ID in the projected database D' for transaction

removal. It leads to hide the sensitive attributes. Line 12 describes the calculation of fitness using the side effects. From line 11 to 16, updating of solutions is described. From line 17 to 25, the fitness values are sorted in the list of descending order. From the list, the solutions with top-half fitness are chosen and these are merged with randomly generated half population of solutions if maximum iterations is not attained in the algorithm. Otherwise, the algorithm will be terminated. In this sROA2RT algorithm, the original database is rescanned using fitness calculation to find the missing attributes and artificial attributes.

Algorithm 1: sROA2RT

Input: D, ϕ , z and η

Output: D' (Sanitized database)

1. Attain H by scanning D.
2. **for** $i \leftarrow 1, m ; l \leftarrow 1, k$ **do**
3. **if** $sl_l \subseteq t_i$ **then**
4. Project t_i from D to form D'
5. **end if**
6. **end for**
7. Initialize the solutions or rain drops from D'.
8. **for** $j \leftarrow 1$ **do**
9. $S_j = \{t_i\}_j, \quad t_i \subseteq D' ; 1 \leq l \leq z$
10. **end for**
11. Generate the number of neighbour points using the condition (8)
12. Determine the fitness for each drop and its neighbour points using (7)
13. Change the current position of drop if it has dominant neighbour point.
14. Set the status of drop as inactive if the drop doesn't have dominant neighbour point after explosion.
15. Generate list of merit order list and remove drops if they have low rank or consider higher neighbour points to drop with high rank.
16. Let t=t+1
17. **if** any active drop is available there and maximum iterations are not attained **then**
18. Arrange a list fitness values in the order of descending
19. Choose top-half fitness from the list
20. Generate half population of solutions randomly from D'
21. Merge solutions from steps 19 and 20 to form new solutions
22. Move to step 11.
23. **else**
24. terminate
25. **end if**

pROA2RT algorithm performs based on the pre-large concept. Using this algorithm, missing and artificial attributes are calculated at each iteration without rescanning the original database. The pre-large attributes (PL) perform like buffers. They reduce the attributes movement from large to small and inverse when the transactions are removed. Algorithm 2 describes the steps of pROA2RT algorithm. Like sROA2RT, pROA2RT algorithm performs. Nevertheless, the generation of pre-large attributes in line 2 reduces the time for rescanning the original database during the process of fitness evaluation. This algorithm leads to reduce the execution time for selecting the optimal solutions.

Algorithm 2: pROA2RT

Input: D, ϕ, z, η, minimum value of minimum support threshold φ_u and pre-large support threshold φ_v

Output: D' (Sanitized database)

1. Set $\varphi_v = \varphi_u \times \left(1 - \frac{z}{|D|}\right)$
2. Attain H and PL using η and φ_v respectively by scanning D.
3. **for** $i \leftarrow 1, m; l \leftarrow 1, k$ **do**
4. **if** $sl_l \subseteq t_i$ **then**
5. Project t_i from D to form D'
6. **end if**
7. **end for**
8. Initialize the solutions or rain drops from D'.
9. **for** $j \leftarrow 1$ **do**
10. $S_j = \{t_i\}_j, \quad t_i \subseteq D'; 1 \le l \le z$
11. **end for**
12. Generate the number of neighbour points using the condition (8)
13. Determine the fitness for each drop and its neighbour points using (7)
14. Change the current position of drop if it has dominant neighbour point.
15. Set the status of drop as inactive if the drop doesn't have dominant neighbour point after explosion.
16. Generate list of merit order list and remove drops if they have low rank or consider higher neighbour points to drop with high rank.
17. Let t=t+1
18. **if** any active drop is available there and maximum iterations are not attained **then**
19. Arrange a list fitness values in the order of descending
20. Choose top-half fitness from the list
21. Generate half population of solutions randomly from D'
22. Merge solutions from steps 20 and 21 to form new solutions
23. Move to step 12.
24. **else**
25. terminate
26. **end if**

4 Results and Discussion

The proposed frameworks are simulated in the platform of python with the system having Intel Core i5 processor and 6GB of memory. These frameworks are evaluated using three databases such as Heart Disease (HD), Heart Attack Prediction (HAP) and Mushroom (Fournier-Viger P et al,2016). HD and HAP are download from the UCI (University of California, Irvine) machine learning repository. Mushroom dataset is the real time dataset related to the gilled mushrooms with 23 species. Besides, these frameworks are analysed in terms of the four side effects and execution time. Besides, the performance of the proposed sROA2RT and pROA2RT is compared with that of sGA2RT and pGA2RT in which GA is used for hiding the data.

4.1. Performance Analysis in Terms of Execution Time

In this section, execution time of the different frameworks is analysed by varying sensitive percentage of frequent attributes using different datasets. Execution time defines the total time to execute the data sanitization scheme. Figure 5 illustrates the execution time analysis using mushroom dataset. As

depicted in the figure, execution time of pGA2RT is reduced to 83% that that of sGA2RT as pGA2RT uses the pre-large concept. However, compared to sGA2RT and pGA2RT, sROA2RT and pROA2RT attain better execution time as the ROA has better convergence speed than GA. Namely, sGA2RT and pGA2RT attains average execution time 31s and 5s respectively while sROA2RT and pROA2RT obtains that 25.25s and 3.57s respectively. The execution time analysis using HD dataset is illustrated in figure 6. As depicted in the figure, execution time of sROA2RT is reduced to 26% than that of sGA2RT. Besides, compared to pGA2RT, execution time of pROA2RT is reduced to 21%. Figure 7 illustrates the execution time analysis using HAP dataset for varying sensitive percentage of frequent attributes. The sGA2RT and pGA2RT attains average execution time 9.5s and 6.5s respectively while sROA2RT and pROA2RT obtains that 7s and 5.25s respectively.

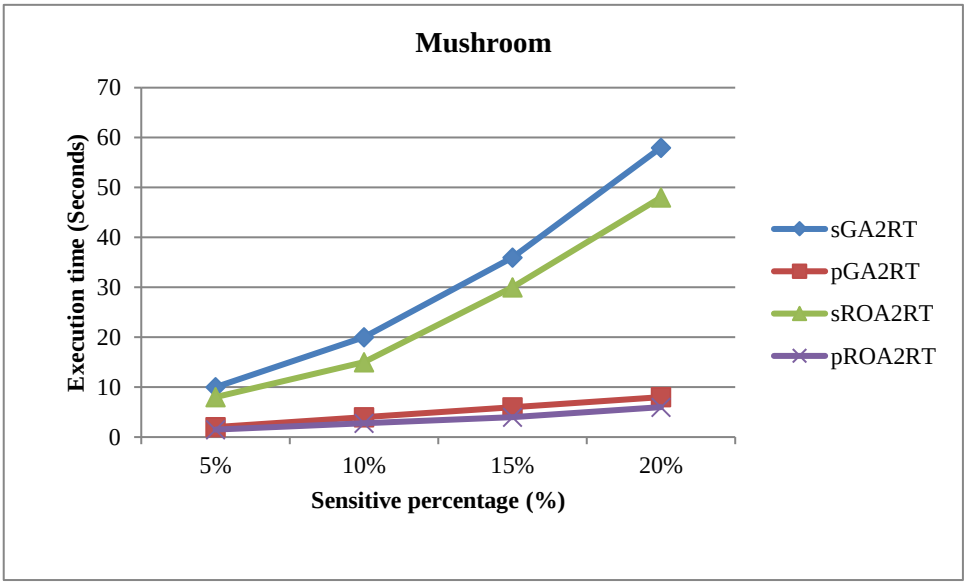

Figure 5: The Execution Time Analysis Using Mushroom Dataset

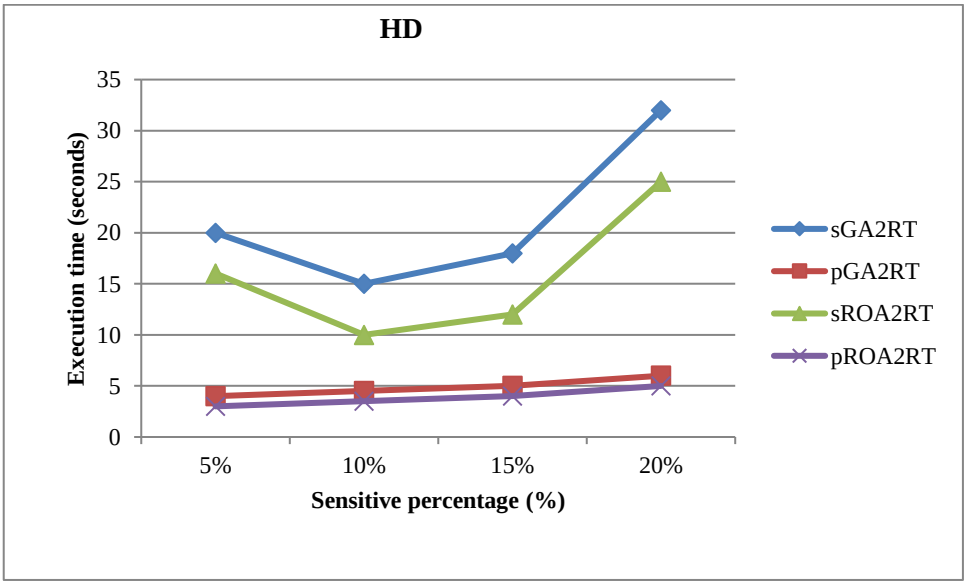

Figure 6: The Execution Time Analysis Using HD Dataset

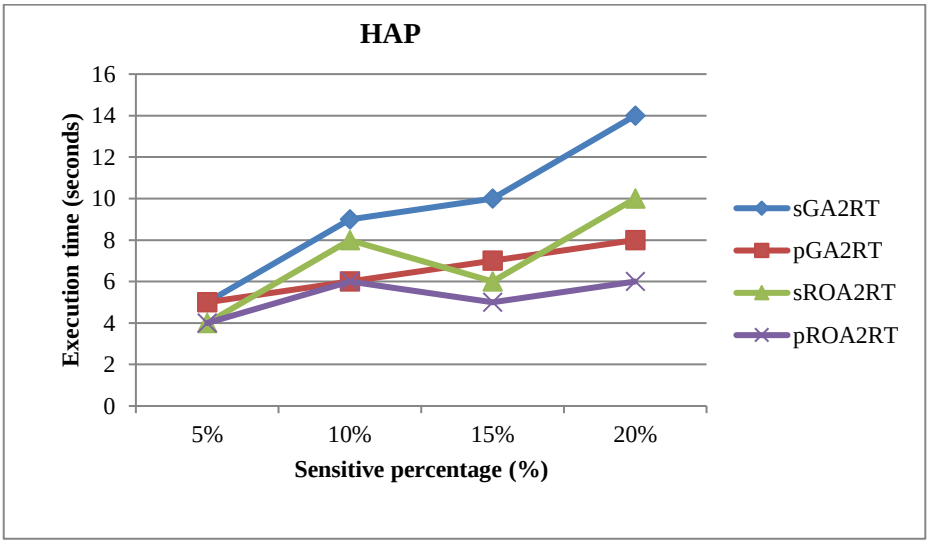

Figure 7: The Execution Time Analysis Using HAP Dataset

4.2. Performance Analysis in Terms of Hiding Failure

In this section, hiding failure of the different frameworks is analysed by varying sensitive percentage of frequent attributes using different datasets. It denotes that the confidential information is supposed to be hidden, despite the fact that it remains after the sanitization process. Figure 8 illustrates the hiding failure analysis using mushroom dataset. As illustrated in the figure, pGA2RT and pROA2RT have fewer hiding failure compared with other two frame works at 10% of sensitive frequent attributes. Also, sROA2RT and pROA2RT have fewer hiding failure than other frameworks at 15% of sensitive frequent attributes. The hiding failure analysis using HD dataset is illustrated in figure 9. From the figure, the reasonable hiding failure is attained for four frameworks when frequent attributes of 5% are used. Figure 10 illustrates the hiding failure analysis using HAP dataset for varying sensitive percentage of frequent attributes. As depicted in the figure, the four frameworks attain reasonable hiding failure while using 5%, 10% and 15% of frequent attributes from HAP dataset.

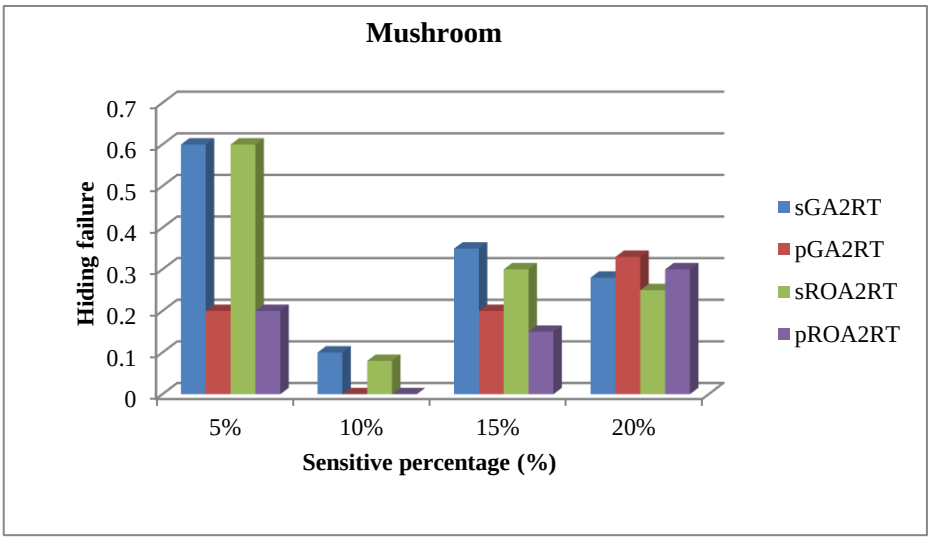

Figure 8: The Hiding Failure Analysis Using Mushroom Dataset

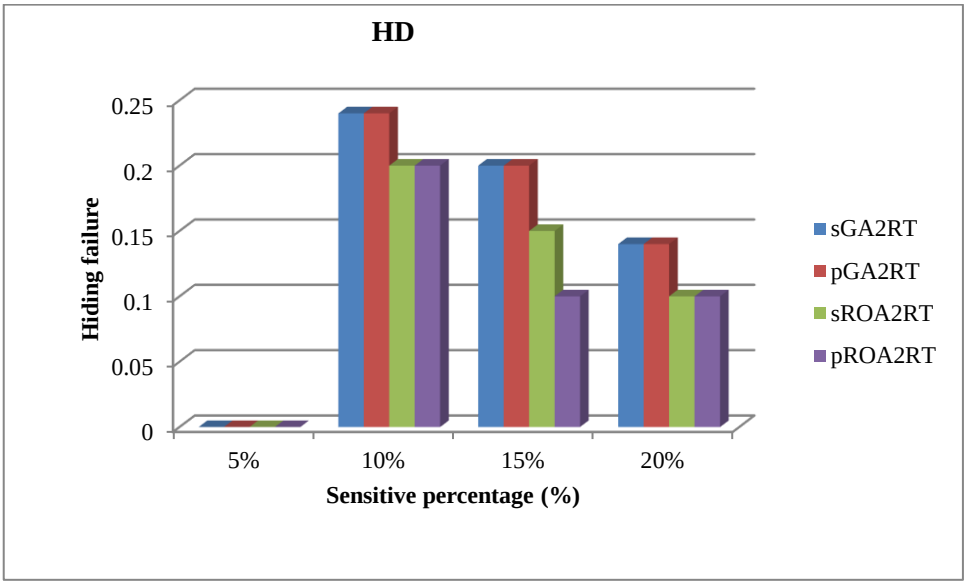

Figure 9: The Hiding Failure Analysis Using HD Dataset

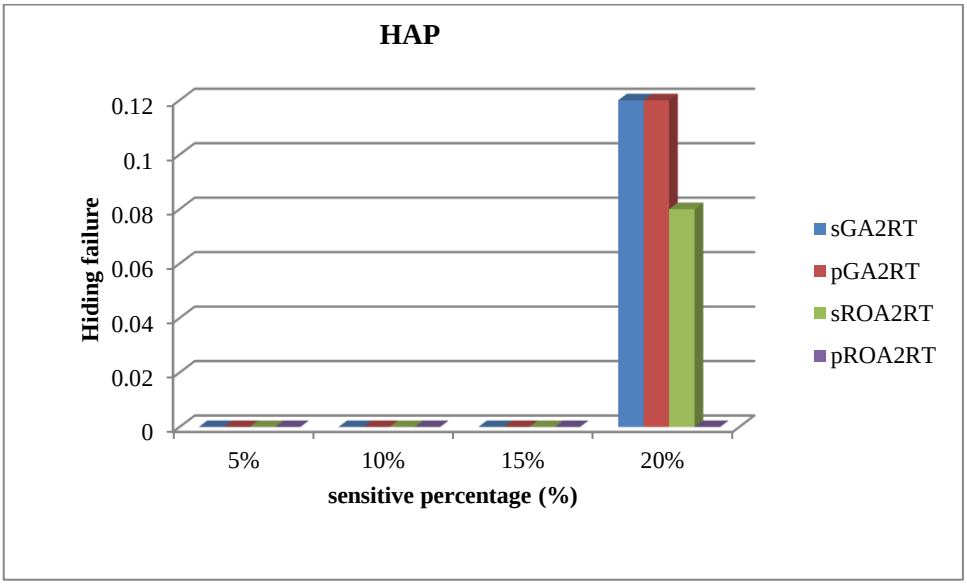

Figure 10: The Hiding Failure Analysis Using HAP Dataset

4.3. Performance Analysis in Terms of Missing Cost

In this section, missing of the different frameworks is analysed by varying sensitive percentage of frequent attributes using different datasets. It implies that the previously discovered information may be overlooked following the sanitization process. Figure 11 illustrates the missing cost analysis using HD dataset. As depicted in the figure, pGA2RT and pROA2RT have fewer missing cost at 5% and 10% of sensitive attributes in the HD dataset. The hiding failure analysis using HAP dataset is illustrated in figure 12. At 5% of frequent attributes, both sGA2RT and pGA2RT have 0.042 of missing cost while sROA2RT and pROA2RT having 0.03 and 0.02 of missing cost respectively. Besides, both sGA2RT and pGA2RT have 0.099 of missing cost while both sROA2RT and pROA2RT having 0.084 of missing cost.

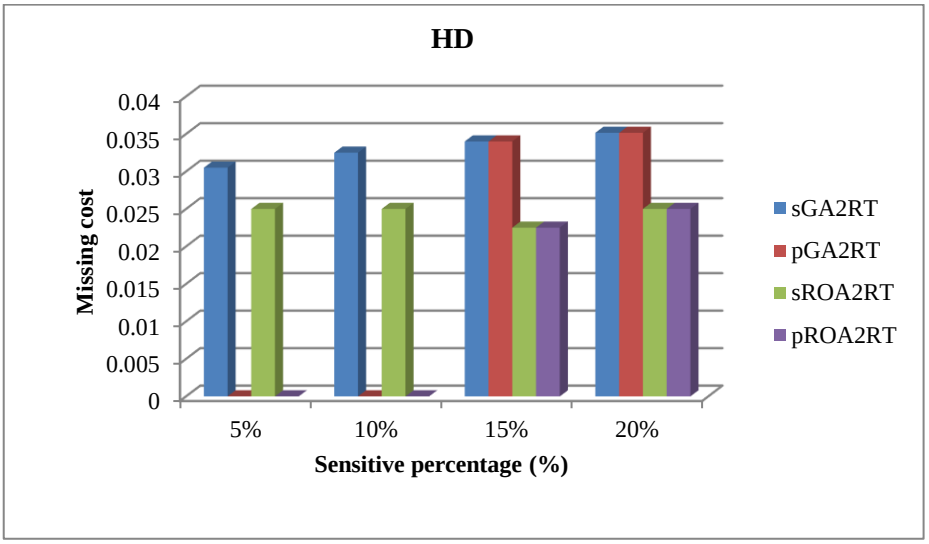

Figure 11: The Missing Cost Analysis Using HD Dataset

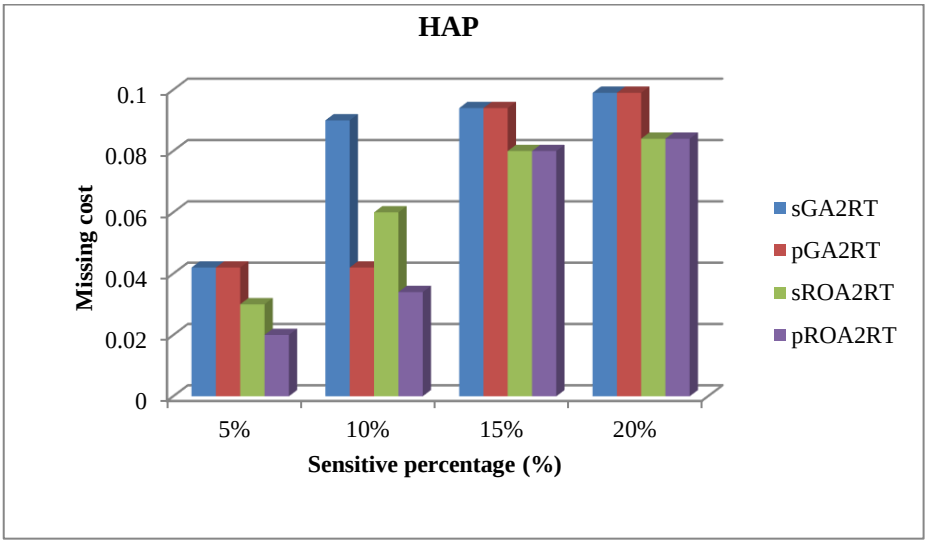

Figure 12: The Missing Cost Analysis Using HAP Dataset

4.4. Performance Analysis in Terms of Dissimilarity

In this section, dissimilarity of the different frameworks is analysed by varying sensitive percentage of frequent attributes using different datasets. It counts the number of deleted transactions between the original and sanitised databases. Figure 13 illustrates the dissimilarity analysis using mushroom dataset. As depicted in the figure, dissimilarity of sGA2RT is reduced to 6% that that of sGA2RT. The sGA2RT and pGA2RT attains average dissimilarity 0.111 and 0.118 respectively while sROA2RT and pROA2RT obtains that 0.092 and 0.097 respectively. The dissimilarity analysis using HD dataset is illustrated in figure 14. As depicted in the figure, dissimilarity of sROA2RT is reduced to 16% than that of sGA2RT. Besides, compared to pGA2RT, dissimilarity of pROA2RT is reduced to 27%. Figure 15 illustrates the dissimilarity analysis using HAP dataset for varying sensitive percentage of frequent attributes. The sGA2RT and pGA2RT attains average dissimilarity 0.01 and 0.009 respectively while sROA2RT and pROA2RT obtains that 0.0089 and 0.0077 respectively.

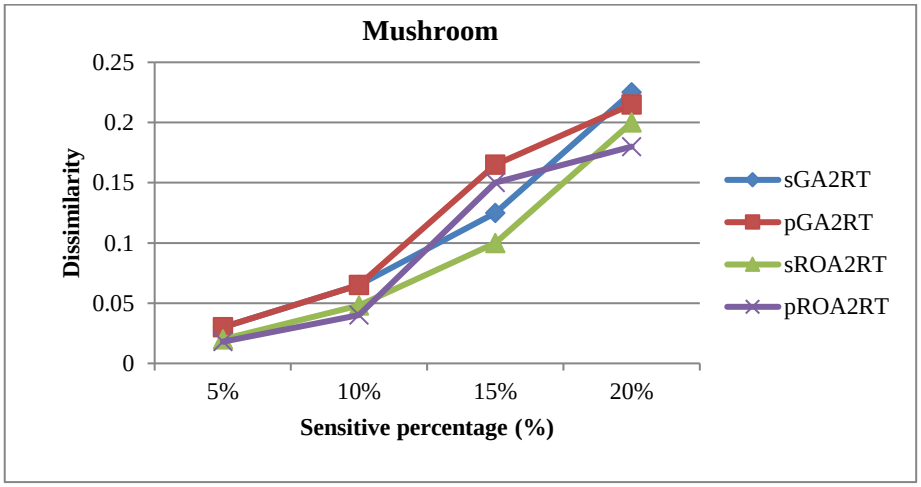

Figure 13: The Dissimilarity Analysis Using Mushroom Dataset

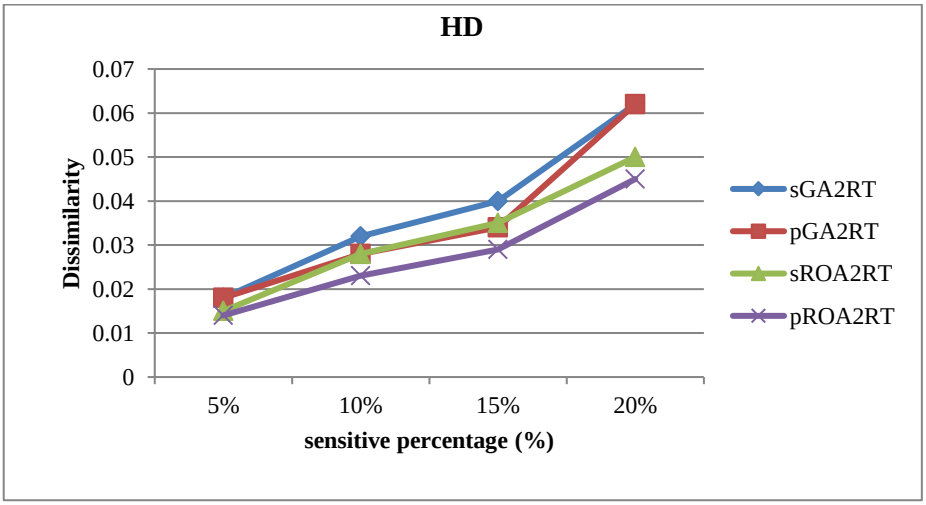

Figure 14: The Dissimilarity Analysis Using HD Dataset

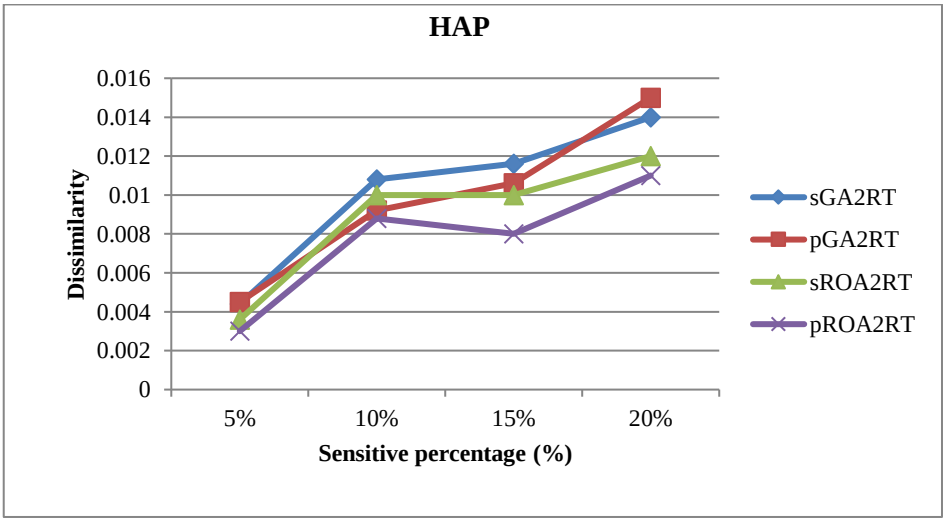

Figure 15: The Dissimilarity Analysis Using HAP Dataset

5 Conclusion

To obtain the reasonable side effects during the process of data sanitization, ROA based two frameworks have been designed in this work. Using sROA2RT and pROA2RT, the suitable transactions to be removed are chosen optimally. The pROA2RT algorithm has been presented to reduce the execution time without rescanning the original database as the sROA2RT algorithm rescans the original database. Along with the side effects such as hiding failure, artificial cost and missing cost, database dissimilarity also has been considered to evaluate the fitness function. The proposed frameworks have been evaluated using HD, HAP and mushroom datasets. Results of the article showed that the proposed sROA2RT and pROA2RT frameworks attained better execution time and side effects than GA based frameworks. In-depth experimental results have shown that the proposed approach can effectively achieve a higher fitness value and attempt to hide a greater amount of sensitive information than previous attempts. As a result, similar health-care systems in the real world are typically quite comprehensive. The two health databases used in our experiments demonstrated that the proposed methods are suitable for use in a clinical setting while also ensuring the security and privacy of patient information. Designing an effective PPDM algorithm for the multi-objective problem is difficult but extremely valuable. In our future research, we will incorporate the multi-threshold model to secure more confidential information for end-users. This issue could also be investigated as a potential future direction for the PPDM research topic.

References

[1] Agrawal, R., Imielinski, T., & Swami, A. (1993). Database mining: A performance perspective. *IEEE transactions on knowledge and data engineering*, *5*(6), 914-925.

[2] Aggarwal, C.C., & Philip, S.Y. (2008). A survey of uncertain data algorithms and applications. *IEEE Transactions on knowledge and data engineering*, *21*(5), 609-623.

[3] Ahmed, U., Srivastava, G., & Lin, J.C.W. (2021). A machine learning model for data sanitization. *Computer Networks*, *189*, 107914.

[4] Chen, M.S., Han, J., & Yu, P.S. (1996). Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and data Engineering*, *8*(6), 866-883.

[5] Dede, H.İ., TİmurkaanC., Kurt, D., & Kofrc, A. (2020). Comparison of Frequent Pattern Mining Algorithms in Internet of Things. *In IEEE 28th Signal Processing and Communications Applications Conference (SIU)*, 1-4.

[6] Evfimievski, A., Srikant, R., Agrawal, R., & Gehrke, J. (2002). Privacy preserving mining of association rules. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 217-228.

[7] Fournier-Viger, P., Lin, J.C.W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., & Lam, H. T. (2016). The SPMF open-source data mining library version 2. *In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, Proceedings, Part III 16*, 36-40. Springer International Publishing

[8] Jangra, S., & Toshniwal, D. (2020). VIDPSO: Victim item deletion based PSO inspired sensitive pattern hiding algorithm for dense datasets. *Information Processing & Management*, *57*(5).

[9] Lin, C.W., Hong, T.P., Yang, K.T., & Wang, S.L. (2015). The GA-based algorithms for optimizing hiding sensitive itemsets through transaction deletion. *Applied Intelligence*, *42*, 210-230.

[10] Lin, J.C.W., Wu, J.M.T., Fournier-Viger, P., Djenouri, Y., Chen, C.H., & Zhang, Y. (2019). A sanitization approach to secure shared data in an IoT environment. *IEEE Access*, *7*, 25359-25368.

[11] Lin, J.C.W., Zhang, Y., Zhang, B., Fournier-Viger, P., & Djenouri, Y. (2019). Hiding sensitive itemsets with multiple objective optimization. *Soft Computing*, *23*, 12779-12797.
[12] Mileva, A., & Tikvesanski, J. (2022). Hiding Data in a Switched Network. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *13*(3), 37-49.
[13] Sharma, U., Toshniwal, D., & Sharma, S. (2020). A sanitization approach for big data with improved data utility. *Applied Intelligence*, *50*, 2025-2039.
[14] Wenzhe, L., Qian, W., Yu, W., Jiadong, R., Yongqiang, C., & Changzhen, H. (2017). Mining Frequent Patterns for Item-Oriented and Customer-Oriented Analysis. *In IEEE 14th Web Information Systems and Applications Conference (WISA)*, 62-67.
[15] Wu, T.Y., Lin, J.C.W., Zhang, Y., & Chen, C.H. (2019). A grid-based swarm intelligence algorithm for privacy-preserving data mining. *Applied Sciences*, *9*(4), 774.
[16] Wu, J.M.T., Srivastava, G., Jolfaei, A., Fournier-Viger, P., & Lin, J.C.W. (2021). Hiding sensitive information in eHealth datasets. *Future Generation Computer Systems*, *117*, 169-180.
[17] Yang, C., Huang, P.C., Lin, Y., Dong, J., Liu, D., Tan, Y., & Liang, L. (2020). Making frequent-pattern mining scalable, efficient, and compact on nonvolatile memories. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *40*(7), 1367-1380.

Authors Biography

M. Madhavi Research Scholar at Annamalai University in the Department of Computer Science and Engineering Chidambaram Tamil Nādu India. Her research interests include Information Security, Cloud Computing, network security, Machine Learning, Optimization techniques in distributed systems, and wireless body sensor networks. M. Madhavi Pursuing Ph.D. in computer science from the Annamalai University Tamil Nādu India. Contact her madhavi.macharapu@gmail.com, Orcid: https://orcid.org/0000-0002-3540-9813

Dr. T.Sasirooba is an Assistant Professor at Annamalai University Chidambaram Tamil Nādu India in the Department of Computer science and Engineering. Her research interests include Image Processing, Information Security, Cloud Computing, network security, Machine Learning, Optimization techniques in distributed systems, and wireless body sensor networks. Dr. T.Sasirooba has a Ph.D. in computer science from the Annamalai University Tamil Nādu India. Contact her at sasirooba@gmail.com, Orcid: https://orcid.org/0000-0001-9544-4496

Dr. G.Kranthi Kumar is a Sr Assistant professor in the department of Computer Science and Engineering at Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India. His research interests include Network Security, Artificial Intelligence and Machine learning, security, and Web services. Dr. G.Kranthi Kumar has a Ph.D. in computer science from Acharya Nagarjuna University Andhra Pradesh India. Contact him at kranthi@vrsiddhartha.ac.in, Orcid: https://orcid.org/0000-0001-8370-8474