Minable Data Publication Based on Sensitive Association Rule Hiding

Fan Yang, Xinyu Lei, Junqing Le, Nankun Mu, and Xiaofeng Liao, *Fellow, IEEE*

***Abstract*—Minable data publication can promote data sharing among commercial companies and further facilitate the development of data-driven services. However, these commercial companies are often reluctant to publish their data due to security concerns. The published data may contain some sensitive information that is minable by malicious entities, leading to data privacy leakage. Therefore, it is highly demanded to develop the technologies supporting minable data publication with privacy protection. In this paper, we propose a privacy-preserved minable data publication scheme (PMDP). PMDP enables selective sensitive association rules hiding while supporting the association rule mining. In PMDP, how to balance the trade-off between data privacy and data utility is the major problem, which can be formulated as a multi-objective optimization problem. To tackle this multi-objective optimization problem, we develop a customized multi-objective evolutionary algorithm (MOEA). In the customized MOEA, the local optimum trapping issue and slow convergence speed issue are hard to be addressed. First, to avoid being trapped into the local optimum, we carefully design a novel mutation method to guarantee the diversity of solutions. Second, to accelerate the convergence speed, we present a preprocessing method before the evolution process of the MOEA. In addition, we introduce the elite learning strategy into the MOEA, so the convergence speed can be further accelerated. At last, experiments are conducted over several datasets to demonstrate the effectiveness of PMDP.**

***Index Terms*—Minable data publication, association rule hiding, multi-objective evolutionary algorithm**

I. Introduction

MINABLE data publication can promote the data sharing among commercial companies and further facilitate the development of data-driven services, such as E-commerce recommendation [1], social network analysis [2], traffic flow prediction [3] and so on. However, these commercial companies often refuse to publish their data due to the security and privacy concerns, because the data may contain some sensitive information. That is to say, there is a conflict between data privacy and data utility in minable data publication. To achieve data privacy in minable data publication, a straightforward solution is to encrypt the data by using standard encryption algorithms (such as AES [4] and DES [5]). However, the formally encrypted data is pseudo-random, and thus, it is hard to mine useful information from the encrypted data. Intuitively, a feasible solution is to weaken the encryption to improve the data utility in minable data publication. Therefore, we aim to develop the technologies supporting some mining functionalities while selectively protecting the sensitive information.

In this paper, we focus on association rule mining, a top 3 algorithm in data mining [6]. Then, we introduce the association rule with a simple example. Suppose there is a dataset about the transactions of a supermarket that has a series of products on sale. Significantly, a data miner can notice that a large part of the transactions containing "Beer" also contains "Diapers". This selection pattern "Beer → Diapers" is an "association rule" that can be discovered by the association rule mining. However, the mining results may lead to privacy and security threats, and the supermarket manager does not want this rule ("Beer → Diapers") to be leaked to the commercial competitors. If it is leaked when the commercial competitors pretend to offer a discount package of buying "Beer" and "Diapers" together, and this may steal a lot of customers from the supermarket manager. Clearly, "Beer → Diapers" is sensitive. In this paper, we select a portion of association rules as the sensitive association rules, the other rules are called non-sensitive rules. To sum up, the target problem is modeled as how to support mining non-sensitive rules while keeping sensitive association rules hidden in minable data publication. For brevity, this problem is named as Sensitive Association Rules Hiding (SARH) problem in this paper.

To address the SARH problem, there are several known methods, and each of them has some limitations. For example, (1) *Low data privacy*. Some works have tackled the SARH problem by decreasing the frequency of sensitive rules. But these works are either weak in the privacy rate [7] or only one sensitive association rule can be hidden at a time [8]. So, data privacy is weak. (2) *Low data utility*. Some works have formulated the SARH problem into a single objective optimization problem [9]. These works only consider the minimization of privacy leakage as the optimization goal and fail to consider how to improve data utility [10]. Hence, data utility is low. (3) *Low efficiency*. Some homomorphic encryption-based methods are presented to address the SARH problem. One of these methods can securely perform data mining over encrypted data [11] and the other allows outsourcing the mining tasks to a third-party cloud [12]. However, the computational costs of these methods are too high to implement on a large dataset.

In this paper, we propose a **p**rivacy-preserved **m**inable **d**ata **p**ublication scheme (PMDP) to enable selective sensitive association rules hiding while support the non-sensitive association

This work is supported in part by National Natural Science Foundation of China (Grant no. 61932006, 61772434, 61806169), in part by National Key R&D Program of China (No. 2018AAA0100101) and in part by Chongqing Technology Innovation and Application Development Project (Grant no. cstc2020jscx-msxmX0156).

Fan Yang, Junqing Le, Nankun Mu and Xiaofeng Liao are with the College of Computer Science, Chongqing University, Chongqing, 400044, P.R. China

Xinyu Lei is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA

Xiaofeng Liao is the corresponding author.

Manuscript received April 19, 2021; revised August 16, 2021.

rules mining. To achieve a controllable balance between data privacy and data utility, PMDP formulates the SARH problem as a multi-objective optimization problem (MOP) with two objectives, i.e., minimizing privacy leakage and maximizing data utility. To efficiently solve the MOP, PMDP develops a customized multi-objective evolutionary algorithm (MOEA). At first, PMDP designs a preprocessing mechanism to filter the irrelevant data so that an amount of irrelevant data can be removed before the evolution of the MOEA. Then, PMDP proposes a novel mutation method consisting of a random and a directional mutation operator in the MOEA to produce the solutions. Furthermore, PMDP introduces the elite learning strategy into the MOEA to accelerate the convergence speed. In addition, to quantify the quality of solutions, PMDP presents a novel fitness function, in which one sub-objective function is used to evaluate the privacy leakage and another sub-objective function is used to evaluate the data utility.

There are two major challenges in PMDP. The first challenge is how to avoid being trapped into the local optimum. The search direction in solution space is greatly influenced by the evolution scheme, and an inappropriate design makes the solutions being trapped into the local optimum [13]. To tackle this challenge, we add a random mutation operator to the mutation, by which a solution will be randomly generated to improve the diversity of solutions. The second challenge is how to accelerate the convergence speed. Ensuring that PMDP can obtain the optimum solution while guarantee the convergence speed is a significant problem for the design of MOEA. Especially, the published dataset may be sparse, which slows down the convergence speed of the MOEA. To tackle this challenge, we present a preprocessing mechanism to reasonably filter the dataset to reduce the search space. In addition, we adopt an elite learning strategy, in which some solutions are generated by learning from the elite set (i.e., the Pareto-optimal solutions) with a certain probability, so the convergence speed can be improved.

The contributions of our work are summarized as follows.

1) We solve the SARH problem in minable data publication by proposing PMDP. In PMDP, the SARH problem is formulated as a MOP, in which minimizing privacy leakage and maximizing data utility are the two objectives. Then, a customized MOEA is designed to solve the MOP, and a controllable balance between data privacy and data utility is realized.
2) A novel mutation method is carefully designed to avoid the solutions of the MOEA being trapped into the local optimum. Specifically, a random mutation operator is added to produce the random solutions and keep the diversity of solutions, so global search ability of MOEA can be improved.
3) A preprocessing mechanism is proposed to reduce the search space, and thus the convergence speed of the MOEA is greatly accelerated. Moreover, the elite learning strategy is introduced to further accelerate the convergence speed of the MOEA.

The rest of this paper is organized as follows. Section II reviews the association rule mining with several necessary definitions, and a formal problem statement is given. Section III gives a detailed description of PMDP. Section IV shows the evaluation results. Section V reviews the related works. Finally, Section VI concludes this paper.

II. Preliminaries And Problem Statement

In this section, we first introduce association rule mining with several necessary definitions. Then, the formal problem statement is given.

A. Association Rules Mining

Consider a transactional dataset $\mathcal{T} = \{t_1, \ldots, t_i, \ldots, t_n\}$, where t_i is a binary vector and denotes the ith transaction. If an item exists in the ith transaction, the corresponding element in t_i equals 1; otherwise, equals 0. All the itemset in $\mathcal{T}$ is $\mathcal{S}$, in which the jth item is denoted by s_j. Suppose that A and B are two subsets of $\mathcal{S}$, and $A \cap B = \emptyset$. Then, a rule can be expressed by $(A \to B)$, where A is called the antecedent and B is called the consequent. The frequently used notations in this paper are summarized in TABLE I.

TABLE I

A Summary of The Frequently Used Notations.

Notation	Meaning
$\mathcal{T} = \{t_i\}$	Dataset, where t_i is the ith transaction
$\mathcal{S} = \{s_j\}$	Itemset, where s_j is the jth item
$minS$	The support threshold
$minC$	The confidence threshold
S_R	The sensitive rules
N_R	The non-sensitive rules
G_R	The ghost rules
C_T	The critical transactions
S_I	The sensitive items
C_R	The critical rules
$\mathbf{x}^{n,i}$	The ith solution of nth iteration
$\mathcal{X}^n = \{\mathbf{x}^{n,i}\}$	The population at nth iteration
$\mathcal{L}$	The Pareto-optimal solutions
$\mathcal{O}^{n,i}$	A learning object that is randomly chosen from $\mathcal{L}$
$\mathcal{P}_m$	Mutation probability
$\mathcal{P}_c$	Crossover probability
$\mathcal{F}(\mathbf{x}^{n,i})$	The fitness value of $\mathbf{x}^{n,i}$
$\mathcal{N}_{pop}$	Maximum population size
$\mathcal{T}_{mut}$	The random mutation condition
$\mathcal{N}_{mut}$	The times that the update is not performed
$\mathcal{N}_{ite}$	The number of iterations
$\mathcal{T}_{fit}$	The terminate condition in terms of fitness
$\mathcal{T}_{ite}$	The terminate condition in terms of iteration

To mine the rules, the following two definitions will be used, i.e., Support and Confidence.

Definition 1 (Support)**.** *The Support of $(A \to B)$ is*

$$Sup_{(A \to B)} = \frac{|A \cup B|}{|\mathcal{T}|} \times 100\%, \qquad (1)$$

where $|\mathcal{T}|$ is the number of transactions, and $|A \cup B|$ is the number of transactions containing both A and B in dataset $\mathcal{T}$.

Definition 2 (Confidence)**.** *The Confidence of $(A \to B)$ is*

$$Conf_{(A \to B)} = \frac{|A \cup B|}{|A|} \times 100\% = \frac{Sup_{(A \to B)}}{Sup_{(A)}}, \qquad (2)$$

where $|A|$ is the number of transactions containing A in dataset $\mathcal{T}$.

For the rule $(A \to B)$, its Support quantifies the occurrence frequency that both A and B exist in the dataset $\mathcal{T}$, and its Confidence quantifies the occurrence frequency that B exists when A exists. Based on the above two definitions, the association rule can be defined as follows.

Definition 3 (Association Rule)**.** *It is said that the association rule $(A \to B)$ holds, if one has*

$$Sup_{(A\to B)} \geq minS \quad and \quad Conf_{(A\to B)} \geq minC, \qquad (3)$$

where $minS$ and $minC$ are the prespecified-thresholds of Support and Confidence, respectively.

The miner can adjust $minS$ and $minC$ to mine useful rules according to their demands.

B. Problem Statement

In this paper, we focus on the SARH problem of minable data publication and aim to present a scheme that can selectively protect sensitive association rules while keeping a high data utility. Based on this, we tackle the SARH problem from the perspective of MOP (the two objects are data privacy and data utility). There have three initial conditions: (i) all rules have already been found from the given dataset by the publisher, (ii) the publisher selects a portion of association rules as the sensitive rules (S_R), and the other rules are called non-sensitive rules (N_R), (iii) the problem is to change the given dataset so that S_R cannot be found and N_R can be found using the same Support and Confidence threshold values. Then the SARH problem can be defined as follows.

Input:

1) Original dataset,
2) Sensitive rules,
3) Two mining thresholds, $minS$ and $minC$.

Output:

1) Hiding schemes.

To address this problem, PMDP is presented.

III. PMDP Design

In this section, we describe the design of PMDP in detail. PMDP consists of four phases, i.e., preprocessing, initialization, evolution, and termination. The flowchart of PMDP is shown in Fig. 1 and its pseudo-code is given as Algorithm 1.

A. Preprocessing

In this subsection, we design a preprocessing mechanism to reduce the search space of the customized MOEA.

Technically, one can hide a rule by reducing its Support or Confidence to below the related mining thresholds. And reducing the Support or Confidence can be achieved by deleting/adding specific items from/into the original dataset. However, these two approaches will cause the following three side-effects.

Algorithm 1: PMDP

```
Input: $\mathcal{T}$, $S_R$, $minS$ and $minC$
Output: $\mathcal{L}$
Generate $C_T$, $S_I$, and $C_R$ // Preprocessing
Generate $\mathcal{X}^0$ and $\mathcal{F}(\mathcal{X}^0)$ // Initialization
$\mathcal{L} \leftarrow \emptyset$, $\mathcal{N}_{mut} \leftarrow 0$, $\mathcal{N}_{fit} \leftarrow 0$
repeat
    $\mathcal{X}^{n+1}$, $\mathcal{L}$, $\mathcal{N}_{mut} \leftarrow$ DiffLearn($\mathcal{X}^n$, $\mathcal{L}$, $\mathcal{N}_{mut}$)
    $\mathcal{N}_{ite} \leftarrow \mathcal{N}_{ite} + 1$
until $\mathcal{T}_{fit} \leq \mathcal{N}_{mut}$ or $\mathcal{T}_{ite} \leq \mathcal{N}_{ite}$;
$\mathcal{L} \leftarrow \{\mathbf{x}^* | \mathcal{F}(\mathbf{x}^*) \preceq \mathcal{F}(\mathbf{x})\}$, $\mathbf{x} \in \{\mathcal{L} \cup \mathcal{X}^{n+1}\}$
```

- **Hiding Failure**. Some S_R exist in the original dataset but are failed to be hidden.
- **Lost Rule**. Some N_R exist in the original dataset but are lost.
- **Ghost Rule** (G_R). Some fake rules do not exist in the original dataset but are generated.

Note that changing the antecedent of S_R results in more serious side-effects than changing the consequent of S_R [14]. Meanwhile, deleting the item will not generate G_R, because the Support of any fake rule can not be increased. Therefore, PMDP decides to delete the consequent of S_R.

Now, we give three definitions for the preprocessing.

Definition 4 (Critical Transaction [10])**.** *A critical transaction is a transaction that supports one or more S_R. Additionally, a transaction is said to support a rule if the transaction contains all the items of this rule.*

Based on Definition 4, only the S_R-related transactions are kept as the critical transactions (C_T), and those un-related transactions can be filtered out to reduce the search space.

Definition 5 (Sensitive Item [10])**.** *If a S_R has more than one item in its consequent, the sensitive item is the item which has the highest frequency in the consequent of S_R and the lowest frequency in N_R. Otherwise, the item in the consequent of S_R will be chosen as the sensitive item.*

PMDP deletes the sensitive item (S_I) to hide S_R. That is to say, only the N_R containing S_I will be affected. Therefore, the critical rule (C_R) is defined so that the lost rule rate can be calculated by evaluating the losses of C_R.

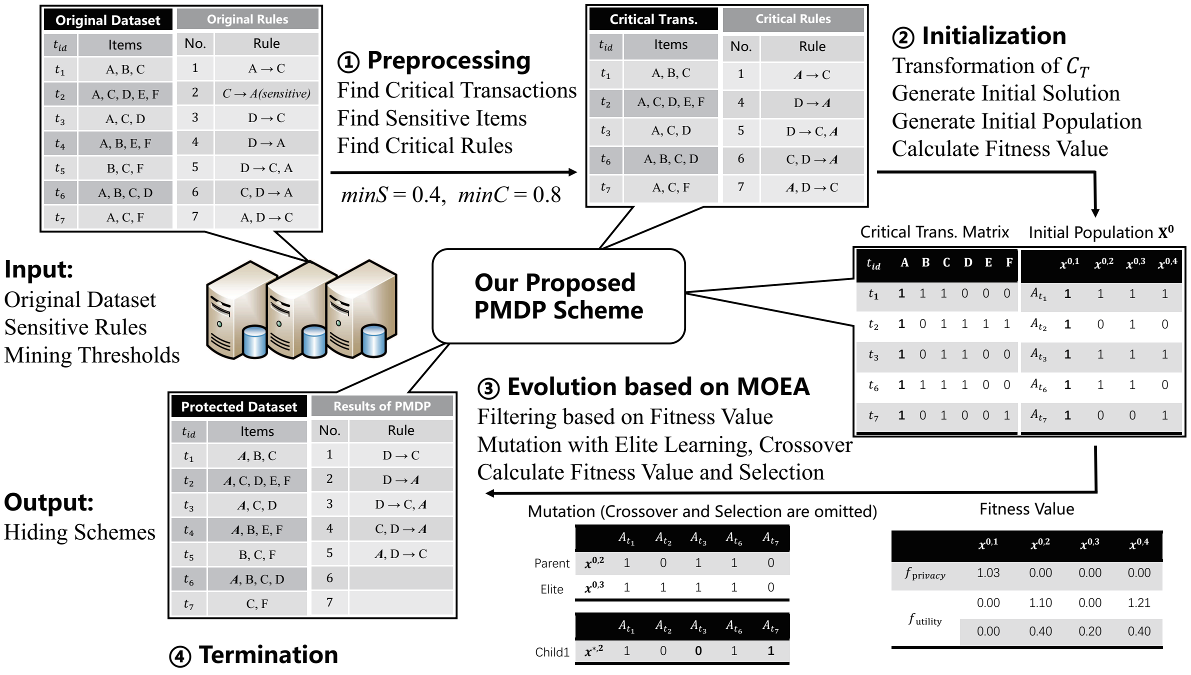
Definition 6 (Critical Rule)**.** *A N_R is designated as a critical rule if it contains any sensitive item.*

The preprocessing mechanism is summarized as follows.

a) PMDP finds C_T,
b) PMDP finds S_I,
c) PMDP finds C_R.

By the above three steps, PMDP can reduce the search space, so the convergence speed can be accelerated. Now, we give a simple example to illustrate preprocessing.

Example. An example dataset is given by Fig. 2(a), which contains 10 transactions and 6 items denoted from a to f. The mined 14 association rules ($minS = 40\%$, $minC = 70\%$) are shown in Fig. 2(b).


① Preprocessing
Find Critical Transactions
Find Sensitive Items
Find Critical Rules
$minS = 0.4,\ minC = 0.8$
② Initialization
Transformation of C_T
Generate Initial Solution
Generate Initial Population
Calculate Fitness Value
Input:
Original Dataset
Sensitive Rules
Mining Thresholds
Our Proposed PMDP Scheme
③ Evolution based on MOEA
Filtering based on Fitness Value
Mutation with Elite Learning, Crossover
Calculate Fitness Value and Selection
Output:
Hiding Schemes
④ Termination
Critical Trans. Matrix
Initial Population $\mathbf{X}^0$
Mutation (Crossover and Selection are omitted)
Fitness Value
Parent
Elite
Child1

Original Dataset		Original Rules	
t_{id}	Items	No.	Rule
t_1	A, B, C	1	A → C
t_2	A, C, D, E, F	2	*C → A(sensitive)*
t_3	A, C, D	3	D → C
t_4	A, B, E, F	4	D → A
t_5	B, C, F	5	D → C, A
t_6	A, B, C, D	6	C, D → A
t_7	A, C, F	7	A, D → C

Critical Trans.		Critical Rules	
t_{id}	Items	No.	Rule
t_1	A, B, C	1	*A* → C
t_2	A, C, D, E, F	4	D → *A*
t_3	A, C, D	5	D → C, *A*
t_6	A, B, C, D	6	C, D → *A*
t_7	A, C, F	7	*A*, D → C

t_{id}	A	B	C	D	E	F		$x^{0,1}$	$x^{0,2}$	$x^{0,3}$	$x^{0,4}$
t_1	**1**	1	1	0	0	0	A_{t_1}	**1**	1	1	1
t_2	**1**	0	1	1	1	1	A_{t_2}	**1**	0	1	0
t_3	**1**	0	1	1	0	0	A_{t_3}	**1**	1	1	1
t_6	**1**	1	1	1	0	0	A_{t_6}	**1**	1	1	0
t_7	**1**	0	1	0	0	1	A_{t_7}	**1**	0	0	1

Protected Dataset		Results of PMDP	
t_{id}	Items	No.	Rule
t_1	*A*, B, C	1	D → C
t_2	*A*, C, D, E, F	2	D → *A*
t_3	*A*, C, D	3	D → C, *A*
t_4	*A*, B, E, F	4	C, D → *A*
t_5	B, C, F	5	*A*, D → C
t_6	*A*, B, C, D	6	
t_7	C, F	7	

		A_{t_1}	A_{t_2}	A_{t_3}	A_{t_6}	A_{t_7}
Parent	$x^{0,2}$	1	0	1	1	0
Elite	$x^{0,3}$	1	1	1	1	0

		A_{t_1}	A_{t_2}	A_{t_3}	A_{t_6}	A_{t_7}
Child1	$x^{*,2}$	1	0	**0**	1	**1**

	$x^{0,1}$	$x^{0,2}$	$x^{0,3}$	$x^{0,4}$
$f_{privacy}$	1.03	0.00	0.00	0.00
f_{utility}	0.00	1.10	0.00	1.21
	0.00	0.40	0.20	0.40

Fig. 1. The flowchart of PMDP.

t_{id}	Items	t_{id}	Items
t_1	a, b, e, f	t_6	a, b, c, e, f
t_2	b, c, d, f	t_7	b, e, f
t_3	b, c, e, f	t_8	a, d, e
t_4	d, f	t_9	a, b, d, e, f
t_5	b, c, d, e	t_{10}	b, d, e, f

(a) Original Dataset

Rules	Sup.	Conf.	Rules	Sup.	Conf.
a → e	40	100	f → e	60	75
c → b	40	100	b → e, f	60	75
b → e	*70*	*87.5*	e → b, f	60	75
b → f	70	87.5	*f → b, e*	*60*	*75*
e → b	70	87.5	b, e → f	60	85.71
f → b	70	87.5	b, f → e	60	85.71
e → f	60	75	e, f → b	60	100

(b) Rules Before Privacy Protection

Fig. 2. The example dataset and the mined rules.

The following two rules are selected as S_R.

$$(i)\, b \rightarrow e \quad \text{and} \quad (ii)\, f \rightarrow b, e$$

Then, the preprocessing is performed as follows.

- Step 1: Find C_T. Based on Definition 4, C_T is constituted by 7 transactions, i.e., $t_1, t_3, t_5, t_6, t_7, t_9$, and t_{10}.
- Step 2: Find S_I. For the first S_R, obviously, item e should be the S_I. For the second S_R, item e satisfies Definition 5. Thus, we have $S_I = \{e\}$.
- Step 3: Find C_R. A N_R containing S_I is identified as a C_R, so there are 9 C_R in the 12 N_R.

t_{id}	Items	t_{id}	Items
t_1	a, *b, e, f*	t_6	a, *b,* c, *e, f*
		t_7	*b, e, f*
t_3	*b,* c, *e, f*		
		t_9	a, *b,* d, *e, f*
t_5	*b,* c, d, *e*	t_{10}	*b,* d, *e, f*

(a) Critical Transactions

Rules	Sup.	Conf.	Rules	Sup.	Conf.
a → *e*	40	100	f → *e*	60	75
			b → *e*, f	60	75
			e → b, f	60	75
e → b	70	87.5	b, *e* → f	60	85.71
			b, f → *e*	60	85.71
e → f	60	75	*e*, f → b	60	100

(b) Critical Rules

Fig. 3. The critical transactions and critical rules obtained by preprocessing.

Fig. 3 shows the results of preprocessing, and we can obtain that $\frac{|C_T|}{|\mathcal{T}|} = \frac{7}{10} = 70\%$ and $\frac{|C_R|}{|N_R|} = \frac{9}{12} = 75\%$. Apparently, the preprocessing can reduce the amount of data and the search space.

B. Initialization

We proceed to introduce the initialization, which is based on a fixed-length representation scheme and can be divided into the following three steps.

a) Transformation. C_T is transformed into a 0-1 matrix. In such matrix, the (i, j)th element equals 1 means that the item s_j exists in transaction t_i. For the matrix of C_T, the size is $|C_T| \times |\mathcal{S}|$, where $|C_T|$ is the number of C_T and $|\mathcal{S}|$ is the number of items.

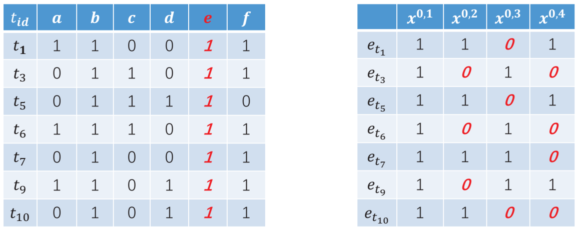
b) Selection. For the matrix of C_T, the S_I-related columns are selected as the initial solution $\mathbf{x}^0$. It is a $|C_T| \times |S_I|$ binary matrix, where $|S_I|$ is the number of S_I.

c) Modification. The elements in $\mathbf{x}^0$ are randomly modified, and then we generate the initial population $\mathcal{X}^0 \in \mathbf{R}^{|C_T| \times |S_I| \times \mathcal{N}_{pop}}$, in which the population size $\mathcal{N}_{pop}$ is empirically set.

Based on the example dataset, we continue to illustrate the initialization.

Example. Fig. 2 shows the itemset $\mathcal{S} = \{a, b, c, d, e, f\}$ and $|\mathcal{S}| = 6$. Then, the initialization is performed as follows.

- Step 1: Transform C_T. Fig. 3(a) shows the C_T. The transformed matrix is shown in Fig. 4(a), in which 4 items (i.e., a, b, e and f) belong to t_1 so it can be represented by 110011. Similarly, t_3, t_5, t_6, t_7, t_9, and t_{10} are respectively denoted by a binary vector.
- Step 2: Generate $\mathbf{x}^{0,1}$. For the given C_T, we pick the S_I-related columns and then obtain the initial solution



t_{id}	a	b	c	d	e	f
t_1	1	1	0	0	1	1
t_3	0	1	1	0	1	1
t_5	0	1	1	1	1	0
t_6	1	1	1	0	1	1
t_7	0	1	0	0	1	1
t_9	1	1	0	1	1	1
t_{10}	0	1	0	1	1	1

(a) The Matrix of Critical Transactions

	$x^{0,1}$	$x^{0,2}$	$x^{0,3}$	$x^{0,4}$
e_{t_1}	1	1	0	1
e_{t_3}	1	0	1	0
e_{t_5}	1	1	0	1
e_{t_6}	1	0	1	0
e_{t_7}	1	1	1	0
e_{t_9}	1	0	1	1
$e_{t_{10}}$	1	1	0	0

(b) Initial Population

Fig. 4. The initialization process of PMDP for the example dataset.

$\mathbf{x}^{0,1}$, which is a 7×1 binary vector since $|C_T| = 7$ and $|S_I| = 1$.

- Step 3: Generate $\mathcal{X}^0$. Several elements of $\mathbf{x}^{0,1}$ are randomly modified to get the initial population $\mathcal{X}^0$. The obtained $\mathcal{X}^0$ has been shown in Fig. 4(b).

C. Evolution

Let us introduce the evolution process of PMDP. Before that, we design the fitness function to quantify the quality of solutions.

1) Fitness Function: The fitness function is represented by

$$\min \mathcal{F}(\mathbf{x}) = (f_{privacy}(\mathbf{x}), f_{utility}(\mathbf{x})), \quad (4)$$

where $f_{privacy}(\mathbf{x})$ is designed to evaluate the data privacy, and $f_{utility}(\mathbf{x})$ is designed to measure the data utility (i.e., the lost rule rate and the ghost rule rate).

Let $|S_R|$, $|C_R|$, $|N_R|$, $|C_T|$, and $|G_R|$ denote the number of S_R, C_R, N_R, C_T, and G_R, respectively. Besides, let R_l denote the lth S_R and R_v denote the vth C_R. Then, for any solution $\mathbf{x}$, the first subfunction $f_{privacy}(\mathbf{x})$ can be represented by

$$f_{privacy}(\mathbf{x}) = \frac{\sum_{l=1}^{|S_R|} r_l + \sum_{l=1}^{|S_R|} h_l}{|S_R|}, \quad (5)$$

where r_l is a binary decision variable. When $r_l = 1$, it indicates that PMDP fails to hide R_l. Besides, we let h_l denote the hiding distance of R_l, and its specific form is

$$h_l = \min\{Conf_{R_l} - minC, Sup_{R_l} - minS\} r_l. \quad (6)$$

In $f_{privacy}(\mathbf{x})$, the value of $\sum_{l=1}^{|S_R|} r_l$ increases by 1 when one S_R is failed to be hidden, which may be caused by the change of multiple transactions. Namely, $\sum_{l=1}^{|S_R|} r_l$ only provides a coarse-grained evaluation about the hiding failure rate. For a fine-grained evaluation, we introduce $\sum_{l=1}^{|S_R|} h_l$, where h_l is calculated by Eq. (6) and can reflect how much at least the Confidence or the Support of R_l needs to be decreased for hiding R_l. Thus, $f_{privacy}(\mathbf{x})$ provides a more precise evaluation of the quality of the solutions on data privacy.

The second subfunction $f_{utility}(\mathbf{x})$ can be represented by

$$f_{utility}(\mathbf{x}) = (\frac{\sum_{v=1}^{|C_R|} p_v + \sum_{v=1}^{|C_R|} l_v}{|N_R|}, \frac{|C_T| - |\mathbf{x}|}{|\mathcal{T}|}), \quad (7)$$

where $|\mathbf{x}|$ is the number of elements with value 1 in $\mathbf{x}$, and p_v is a binary decision variable. When $p_v = 1$, it indicates that PMDP loses R_v. Besides, let l_v denote the lost distance of R_v, and its specific form is

$$l_v = \max\{minC - Conf_{R_v}, minS - Sup_{R_v}\} p_v. \quad (8)$$

In $f_{utility}(\mathbf{x})$, the value of $\sum_{v=1}^{|C_R|} p_v$ increases by 1 when a C_R loses during the hiding process. Similar to $f_{privacy}(\mathbf{x})$, $\sum_{v=1}^{|C_R|} p_v$ is a coarse-grained evaluation about the lost rule rate. Thus, $\sum_{v=1}^{|C_R|} l_v$ is adopted as a fine-grained evaluation, where l_v is calculated by Eq. (8) and can reflect how much at most the Confidence or the Support of R_v needs to be increased for avoiding the loss of R_v. Owing to that PMDP hides S_R by removing its consequent, there is no G_R generated. Hence, the ghost rule rate of PMDP identically equals 0. Besides, we hope to further reduce the loss of data utility, so we introduce the data change rate $\frac{|C_T| - |\mathbf{x}|}{|\mathcal{T}|}$ as the auxiliary information to quantify the quality of solutions.

During the evolution, PMDP calculates the fitness of each solution and finds the Pareto-optimal solutions. Fig. 5 shows the relationship between the general optimal solution and the Pareto-optimal solutions. Pareto-optimal means that it is impossible to make at least one object better without making any other object's value worse.

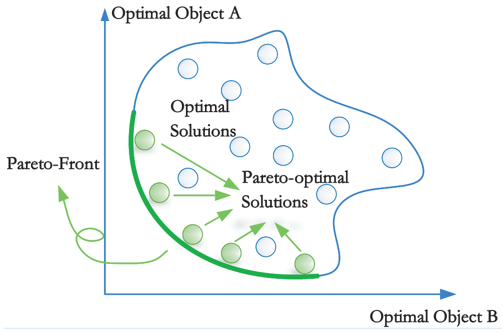


Fig. 5. The relationship between the general optimal solution and the Pareto-optimal solutions.

Then, we give the following definitions of Dominate and Pareto-optimal.

Definition 7 (Dominate [15])**.** *A vector* $\mathcal{F}^* = (f_1^*, \ldots, f_d^*)^\top$ *is said to dominate another vector* $\mathcal{F} = (f_1, \ldots, f_d)^\top$, *if*

$$\mathcal{F}^* \preceq \mathcal{F}$$

holds, where $\mathcal{F}^* \preceq \mathcal{F}$ *means that* $\forall\, i \in \{1, \ldots, d\}, f_i^* \leq f_i$ *and* $\exists\, i \in \{1, \ldots, d\}, f_i^* < f_i$.

Definition 8 (Pareto-optimal [15])**.** *A solution* $\mathbf{x}^*$ *is called a Pareto-optimal if no other solutions dominate it.*

2) Evolution with Elite Learning: We design an evolution scheme DiffLearn with an elite learning strategy. DiffLearn includes filtering, mutation, crossover, and selection. The pseudo-code is given as Algorithm 2, and the main steps of DiffLearn are described as follows.

a) Filtering. The Filtering is performed as

$$\mathcal{L} = \{\mathbf{x}^* | \mathcal{F}(\mathbf{x}^*) \preceq \mathcal{F}(\mathbf{x})\}, \quad (9)$$

```
Algorithm 2: DiffLearn
Input: $\mathcal{X}^n$, $\mathcal{L}$ and $\mathcal{N}_{mut}$
Output: $\mathcal{X}^{n+1}$, $\mathcal{L}$ and $\mathcal{N}_{mut}$
$\mathcal{L} \leftarrow \{\mathbf{x}^* | \mathcal{F}(\mathbf{x}^*) \preceq \mathcal{F}(\mathbf{x})\}$ // Filtering
for $i = 1, 2, ..., \mathcal{N}_{pop}$ do
    Select $\mathcal{O}^{n,i}$ from $\mathcal{L}$ // Select learning
        object
    if $\mathcal{N}_{mut} > \mathcal{T}_{mut}$ then
        $\mathbf{d}^{*,i} \leftarrow \mathcal{M}^{rand}$;
    else
        $\mathbf{d}^{*,i} \leftarrow \lceil \mathcal{P}_m - r_1 \rceil \mathcal{O}^{n,i} + \lceil r_1 - \mathcal{P}_m \rceil$
            // Mutation
        $\mathbf{x}^{*,i} \leftarrow \lceil \mathcal{P}_c - r_2 \rceil \mathbf{d}^{*,i} + \lceil r_2 - \mathcal{P}_c \rceil \mathbf{x}^{n,i}$
            // Crossover
        Calculate $\mathcal{F}(\mathbf{x}^{*,i})$ // Calculate fitness
        if $\mathcal{F}(\mathbf{x}^{*,i}) \preceq \mathcal{F}(\mathbf{x}^{n,i})$ then
            $\mathbf{x}^{n+1,i} \leftarrow \mathbf{x}^{*,i}$; $\mathcal{F}(\mathbf{x}^{n+1,i}) \leftarrow \mathcal{F}(\mathbf{x}^{*,i})$
                // Selection
            $\mathcal{N}_{mut} \leftarrow 0$;
        else
            $\mathcal{N}_{mut} \leftarrow \mathcal{N}_{mut} + 1$;
```

where $\mathbf{x}^*$ is a Pareto-optimal solution that satisfies Definition 8, and the Pareto-optimal solutions constitute the elite set $\mathcal{L}$. Both $\mathbf{x}^*$ and $\mathbf{x}$ belong to $\{\mathcal{L} \cup \mathcal{X}^n\}$.

b) Mutation. Set a mutation threshold $\mathcal{T}_{mut}$ smaller than the terminate threshold $\mathcal{T}_{fit}$ (Sec. III-D). Besides, we let $\mathcal{N}_{mut}$ denote the times that the update is not performed. Then, the Mutation is performed as follows.

(i) $\mathcal{N}_{mut} > \mathcal{T}_{mut}$,

$$\mathbf{d}^{*,i} = \mathcal{M}^{rand}, \tag{10}$$

where $\mathcal{M}^{rand}$ is a solution randomly generated and $\mathbf{d}^{*,i}$ is utilized to generate the ith solution of the next iteration.

(ii) $\mathcal{N}_{mut} \leq \mathcal{T}_{mut}$,

$$d_{k,m}^{*,i} = \begin{cases} o_{k,m}^{n,i}, & \text{if } r_1 \leq \mathcal{P}_m, \\ 1, & \text{otherwise,} \end{cases} \tag{11}$$

where the subscript (k, m) denotes the (k, m) element of a variable, and $k \in \{1, \ldots, |C_T|\}$, $m \in \{1, \ldots, |S_I|\}$, hereinafter are the same. Hence, $d_{k,m}^{*,i}$ is the (k, m)th element of $\mathbf{d}^{*,i}$, and $o_{k,m}^{n,i}$ is the (k, m)th element of $\mathcal{O}^{n,i}$, a learning object randomly chosen from the elite set $\mathcal{L}$. Moreover, $\mathcal{P}_m$ denotes the mutation probability, and r_1 is uniformly distributed in (0, 1).

c) Crossover. The Crossover is performed as

$$x_{k,m}^{*,i} = \begin{cases} d_{k,m}^{*,i}, & \text{if } r_2 \leq \mathcal{P}_c, \\ x_{k,m}^{n,i}, & \text{otherwise,} \end{cases} \tag{12}$$

where $x_{k,m}^{*,i}$ is an element of $\mathbf{x}^{*,i}$ that denotes the ith solution of each iteration. The crossover probability is denoted by $\mathcal{P}_c$. The random number r_2 is uniformly distributed in (0, 1).

d) Selection. The Selection is performed as

$$\mathbf{x}^{n+1,i} = \begin{cases} \mathbf{x}^{*,i}, & \text{if } \mathcal{F}(\mathbf{x}^{*,i}) \preceq \mathcal{F}(\mathbf{x}^{n,i}), \\ \mathbf{x}^{n,i}, & \text{otherwise,} \end{cases} \tag{13}$$

where $\mathbf{x}^{n+1,i}$ is the ith solution of (n+1)th iteration.

Take $\mathbf{x}^{n,i}$ as an example, the specific evolution process is as follows.

- Step 1: Before the evolution of the nth iteration, DiffLearn first performs Filtering and then the obtained Pareto-optimal solutions are selected as the elite set $\mathcal{L}$. For the 1st iteration, the elite set $\mathcal{L}$ is generated by performing the Filtering on the initial population, and each element of $\mathcal{L}$ is an elite solution.
- Step 2: DiffLearn checks whether random mutation condition $\mathcal{N}_{mut} > \mathcal{T}_{mut}$ holds or not. If the condition holds, a random solution $\mathcal{M}^{rand}$ will be generated and assigned to $\mathbf{d}^{*,i}$ by Eq. (10), and then jump directly to Step 3. If not, DiffLearn will perform Mutation for each element of $\mathbf{x}^{n,i}$ as Eq. (11). When r_1 is smaller than $\mathcal{P}_m$, the value $o_{k,m}^{n,i}$ of the selected learning object $\mathcal{O}^{n,i}$ will be assigned to $d_{k,m}^{*,i}$. Otherwise, the value of $d_{k,m}^{*,i}$ will be set to 1.
- Step 3: DiffLearn performs as Eq. (12). If the random number r_2 is smaller than $\mathcal{P}_c$, the mutation result $d_{k,m}^{*,i}$ will be assigned to $x_{k,m}^{*,i}$. If not, DiffLearn keeps the parent solution's value, i.e., $x_{k,m}^{*,i} = x_{k,m}^{n,i}$. Then, the child solution $\mathbf{x}^{*,i}$ is generated.
- Step 4: As Eq. (13), PMDP first calculates the child solution's fitness. Then, the solution with smaller fitness will be kept as the parent solution of next iteration. If there is no update during this iteration, the value of $\mathcal{N}_{mut}$ is incremented by 1.

Compared with the general EA in [16], DiffLearn designs an extra step, Filtering (Step 1), in which the elite set is composed of the Pareto-optimal solutions. The convergence speed of PMDP can be improved with the elite learning strategy. Some researchers ignored that the utilization of elite learning in evolution [17]. Besides, DiffLearn will assign 1 to $d_{k,m}^{*,i}$ if r_1 is greater than the mutation probability $\mathcal{P}_m$. It is different from the mutation which uses the exclusive or operation to get the mutation result [18]. This kind of mutation may cause a lot of unnecessary change. From Definition 3, we know that the fewer change means the smaller of lost rule rate. In addition, to avoid trapping at a local optimum, PMDP introduces $\mathcal{N}_{mut}$. If $\mathcal{N}_{mut}$ exceeds a given threshold, DiffLearn will generate a random solution, which can improve the population diversity.

D. Termination

Two parameters, $\mathcal{T}_{fit}$ and $\mathcal{T}_{ite}$, are selected as the terminate conditions. Once $\mathcal{T}_{fit} \leq \mathcal{N}_{mut}$ holds, the evolution terminates and arrives at the final result. Besides, $\mathcal{N}_{ite}$ denotes the number of iterations, and the evolution terminates when $\mathcal{N}_{ite}$ exceeds $\mathcal{T}_{ite}$. Hence, the terminate condition is $\mathcal{T}_{ite} \leq \mathcal{N}_{ite}$ or $\mathcal{T}_{fit} \leq \mathcal{N}_{mut}$. The evolution process of DiffLearn will repeat until at least one of these two termination conditions is satisfied.

IV. Performance Evaluation

In this section, experiments are conducted to evaluate the performance of PMDP. We repeat the experiment 10 times, and the averages of these experimental results are compared with COA4ARH [10]. The experimental-related settings are given as follows.

- **Implementation.** PMDP is implemented by MATLAB. Experiments are performed on a system with Microsoft Windows 7 32-bit Operating System (3.30GHz Intel Core i5-4590 processor CPU and 4.00 GB RAM).
- **Datasets.** We adopt three typically public datasets[1], i.e., Mushroom, c20d10k, and BMS1. The characteristics of the three datasets are shown in TABLE II, where **Avg.Len** and **Max.Len** denote the average and the maximum length of the transactions, respectively.

TABLE II
Characteristics of Datasets

Dataset	Trans.	Items	Avg.Len	Max.Len	Type
Mushroom	8,416	119	23	23	Dense
c20d10k	10,000	192	20	20	Dense
BMS1	59,601	497	2.51	267	Sparse

- **Parameters Setting.** The parameters of the three datasets are shown in TABLE III, in which $\mathcal{N}_{pop}$ can be empirically set by users according to their demands.

TABLE III
Parameters Setting of Datasets

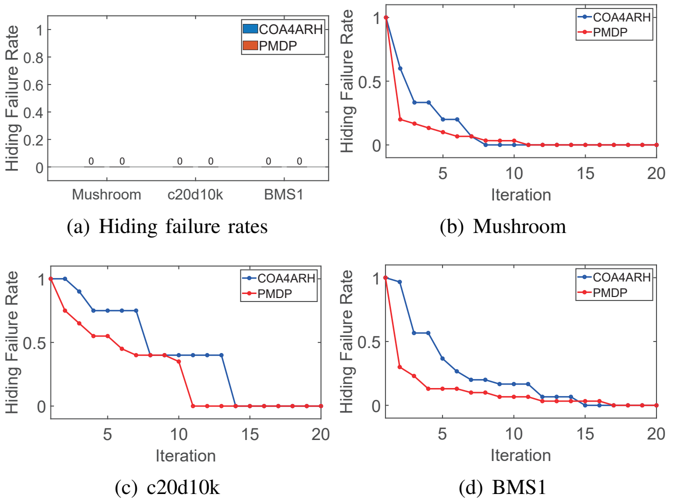
Dataset	Rules	$minS$	$minC$	$\mathcal{P}_m$	$\mathcal{P}_c$	$\lvert S_R \rvert$	$\mathcal{T}_{ite}$	$\mathcal{T}_{fit}$	$\mathcal{N}_{pop}$
Mushroom	3,828	40%	70%	0.8	0.9	30	20	5	100
c20d10k	91,878	50%	70%	0.6	0.8				
BMS1	19,655	0.1%	5%	0.5	0.7				

- **Metrics.** The main used metrics are the three side-effects, i.e., (1) hiding failure rate, (2) lost rule rate, and (3) ghost rule rate. Besides, another two metrics are used, i.e., (1) the running time, and (2) the ratio of $\frac{|C_T|}{|\mathcal{T}|}$ and $\frac{|C_R|}{|N_R|}$.

A. Performance of Data Privacy

We first evaluate the performance of data privacy, i.e., the hiding failure rate. The hiding failure rates of PMDP are compared with that of COA4ARH, and they are shown in Fig. 6. To conduct the comparison, we choose the solution which has the lowest hiding failure rate at each iteration. The detailed analysis is given as follows.

Fig. 6 shows the final results and the evolution process of PMDP and COA4ARH. From Fig. 6(a), we can see that both PMDP and COA4ARH converge to 0. From Fig. 6(b), Fig. 6(c), and Fig. 6(d), one can observe that PMDP shows a faster convergence speed than COA4ARH for the three datasets. To sum up, PMDP ensures an optimal solution and provides a faster convergence speed.



(a) Hiding failure rates (b) Mushroom

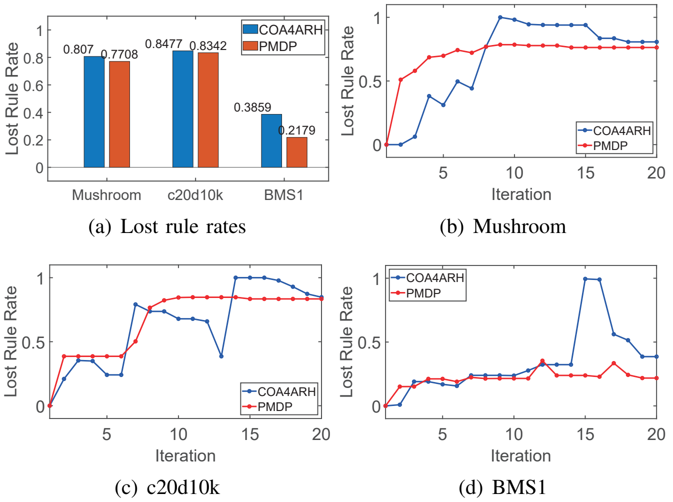
(c) c20d10k (d) BMS1

Fig. 6. Hiding failure rate comparison of PMDP and COA4ARH under three datasets.

B. Performance of Data Utility

We then evaluate the performance of data utility, i.e., the lost rule rate and the ghost rule rate. The detailed analysis is given as follows.

1) Evaluation of the lost rule rate: For the evaluation of the lost rule rates that obtained by PMDP and COA4ARH, we give the related results in TABLE IV, where **Mean** and **Std.** denote the mean and the standard deviation of the lost rule rates at 10 times running, respectively. Fig. 7 shows the mean of the final lost rule rates and the related evolution process of PMDP and COA4ARH, respectively. Specifically, we select the solution which has the lowest hiding failure rate at each iteration for this comparison. If some solutions have the same hiding failure rate, we select the solution which has the lowest lost rule rate.



(a) Lost rule rates (b) Mushroom

(c) c20d10k (d) BMS1

Fig. 7. Lost rule rate comparison of PMDP and COA4ARH under three datasets.

Specifically, the final lost rule rates obtained by PMDP and COA4ARH are plotted in Fig. 7(a). For PMDP, the lost rule rates of the three datasets are 77.08%, 83.42%, and 21.79%,

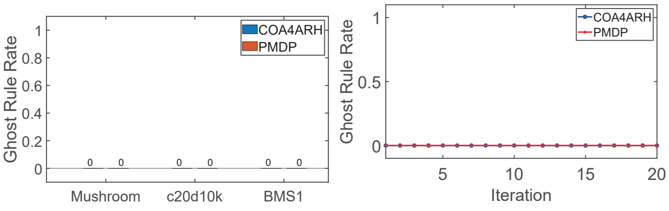
[1] http://www.philippe-fournier-viger.com/spmf/

TABLE IV
THE STATISTICS INFORMATION OF THE LOST RULE RATE ($\frac{|LostRules|}{|C_R|}$)

Dataset	Methods	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9	No.10	Mean	Std.
Mushroom	PMDP	0.8058	0.7775	0.8058	0.8054	0.7487	0.8054	0.7413	0.7506	0.7340	0.7336	0.7708	0.0323
	COA4ARH	0.7871	0.8009	0.7914	0.8154	0.8851	0.7956	0.8009	0.7871	0.7914	0.8154	0.8070	0.0292
c20d10k	PMDP	0.8236	0.8253	0.8123	0.8441	0.8441	0.8815	0.8123	0.8123	0.8343	0.8518	0.8342	0.0220
	COA4ARH	0.8074	0.8890	0.8332	0.8342	0.8748	0.8890	0.8074	0.8332	0.8342	0.8748	0.8477	0.0315
BMS1	PMDP	0.2319	0.1780	0.2696	0.1529	0.2264	0.2201	0.2592	0.1931	0.2293	0.2182	0.2179	0.0353
	COA4ARH	0.3338	0.3744	0.4342	0.4342	0.3684	0.3778	0.3934	0.3338	0.3744	0.4342	0.3859	0.0382

respectively. For COA4ARH, the lost rule rates of the three datasets are 80.70%, 84.77%, and 38.59%, respectively. We can see that the lost rule rate of COA4ARH is higher than that of PMDP. Besides, Fig. 7(b) shows the evolution process of the lost rule rates obtained by PMDP and COA4ARH under the Mushroom dataset. We can see that PMDP shows a faster convergence speed than COA4ARH, and we can obtain this conclusion from Fig. 7(c) and Fig. 7(d) as well. From this point of view, the concept of C_R contributes to the acceleration of the convergence speed. To sum up, PMDP has better performance on data utility (i.e., lower lost rule rate) than COA4ARH.

2) Evaluation of the ghost rule rate: This part evaluates the ghost rule rates of PMDP and COA4ARH.



(a) Ghost rule rates (b) The evolution of ghost rule rate

Fig. 8. Ghost rule rate comparison of PMDP and COA4ARH under three datasets.

Fig. 8 shows that the ghost rule rate identically equals 0 in the hiding process of PMDP and COA4ARH. The ghost rule is a common issue for several existing algorithms, such as DSR [7]. Apparently, the influence of ghost rule has been successfully eliminated in PMDP.

C. Performance on Optional Hiding

We now turn to evaluate the performance on optional hiding, namely, the selectivity of the solution. Fig. 9 shows the Pareto-optimal solutions obtained by PMDP and the optimal solution obtained by COA4ARH under three datasets.

Fig. 9 shows that the solutions of PMDP are almost evenly distributed on the solution space. The reason is that PMDP considers the importance of each side-effect equally. So, there are Pareto-solutions generated. Each Pareto-optimal solution of PMDP is an optional hiding scheme for a dataset to deal with the SARH problem. Meanwhile, COA4ARH chooses to prioritize the hiding failure rate, thus, there is only one solution obtained by COA4ARH, which makes the publisher have no other choices. To sum up, PMDP provides a variety of options for the data publisher to achieve the selectively S_R hiding.

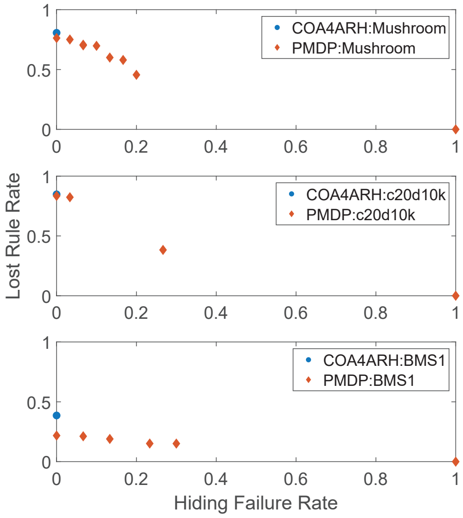


Fig. 9. Performance on optional solutions under three datasets.

Besides, PMDP formulates the SARH problem as a MOP, in which one objective is data privacy (i.e., hiding failure rate), and another objective is data utility (i.e., lost rules rate). From Fig. 9, one can observe that the hiding failure rate is negatively correlated with the lost rule rate. The results of Pareto-optimal solutions demonstrate that it is reasonable to adopt the MOEA.

D. Performance on Efficiency

In this section, we demonstrate the efficiency of PMDP.

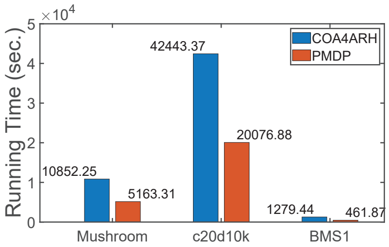
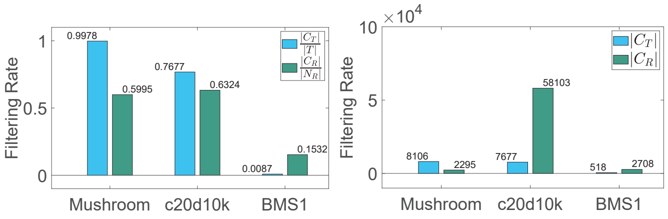


Fig. 10. Running time comparison of PMDP and COA4ARH under three datasets.

Fig. 10 shows the running time of PMDP and COA4ARH under three datasets. One can observe that the running time of PMDP under three datasets is about half of the running time of COA4ARH. In addition, c20d10k costs more time than Mushroom and BMS1.



(a) $\frac{|C_T|}{|\mathcal{T}|}$ and $\frac{|C_R|}{|N_R|}$

(b) $|C_T|$ and $|C_R|$

Fig. 11. Comparison of $\frac{|C_T|}{|\mathcal{T}|}$ and $\frac{|C_R|}{|N_R|}$ under three datasets.

Fig. 11 shows $\frac{|C_T|}{|\mathcal{T}|}$ (light blue) and $\frac{|C_R|}{|N_R|}$ (green) of the three datasets, and it demonstrates the rationality of the proposed preprocessing mechanism again. One can observe that the $\frac{|C_T|}{|\mathcal{T}|}$ of the first two datasets are significantly high and even approach 1. For BMS1 dataset, $\frac{|C_T|}{|\mathcal{T}|}$ is significantly low and nearly equals 0. Besides, the $\frac{|C_R|}{|N_R|}$ of the first two datasets is relatively high, and both of them are over 50%. Meanwhile, the $\frac{|C_R|}{|N_R|}$ of BMS1 is 15.32%. Because we only need to calculate C_R for evaluating the lost rule rate, the computational cost of PMDP is reduced. Apparently, PMDP reduces the search space of the customized MOEA in dealing with the SARH problem.

From Fig. 10 and Fig. 11, we can infer that the running time may be influenced by the type of dataset, $\frac{|C_T|}{|\mathcal{T}|}$, and $\frac{|C_R|}{|N_R|}$. Especially, higher $\frac{|C_R|}{|N_R|}$ may cause a longer running time. To sum up, PMDP shows better performance on efficiency than COA4ARH.

V. Related Work

This section briefly reviews the work related to this paper. To our knowledge, these methods can be roughly divided into two types, i.e., the traditional method and the heuristic method.

A. Traditional Method

1) Reconstruction-based Method: The reconstruction-based method first reconstructs the dataset and then conducts the association rules mining, such that the privacy can be preserved. For instance, a reconstruction approach is presented to estimate the distribution of data, in which several classifiers are further designed to evaluate the performance of the reconstructed data and the original data [19]. Besides, to balance the privacy and the data utility, another reconstruction-based algorithm called DR-PPFIM is proposed in [20]. DR-PPFIM first identifies the frequent itemsets related to sensitive frequent itemsets and removes them, and then a reconstruction scheme is performed. But, the adopted reconstruction scheme may not be global optimum for the hiding purpose.

2) Sanitization-based Method: To protect sensitive information, the sanitization-based method has emerged [21]. For instance, some malicious attackers can utilize social media information in a published dataset to predict private information. To reduce the accuracy of this kind of attack, a sanitization-based method is proposed in [22]. Besides, the data sharing among a variety of organizations also causes privacy problems. Then, the researchers design three sanitization-based mining algorithms for privacy-preserving utility mining [23]. Our proposed scheme hides the sensitive rules by deleting the specific items from the dataset, which is similar to the sanitization-based method. But, the proposed scheme provides more choices and a data utility guarantee.

B. Heuristic method

1) SOEA-based method: The single-objective evolutionary algorithm (SOEA)-based method shows great performance on the finding of the optimal solution [24], [25]. For example, owing to the great ability to explore the search space, the particle swarm optimization (PSO) algorithm is applied to hide rules [26]. Compared with the genetic algorithm (GA)-based method [27] which pre-defines weights for those side-effects, the PSO algorithm can outperform on the effectiveness of protecting/hiding S_R. In addition, a cuckoo search optimization algorithm is adopted for deleting/inserting items from/into the dataset, and a solution with the fewest side-effects is obtained [10]. It shows better performance than the GA-based hiding strategy [28] and DSR [7]. But these works only consider the minimization of privacy leakage as the optimization goal and fail to consider improving data utility. Hence, the data utility of the protected dataset is low.

2) MOEA-based Method: Several MOEA-based method are presented to solve the problem that exists in the SOEA-based method [29]–[31]. In the beginning, the multi-objective genetic algorithm is utilized to mine association rule [29]. Some works have aimed at association rules mining, such as a MOEA-based method is presented for mining the fuzzy emerging patterns [32], another indexed set representation scheme-based MOEA for mining high utility patterns [33], and a hybrid MOEA methods for rapidly and directly mining the high-quality rules [34]. Recently, the researchers focus on the MOEA-based method to privacy-preserved mine association rules. For instance, a MOEA-based algorithm is presented, but it can hide only one sensitive association rule at a time [8]. Based on MOEA, another method of frequent itemset hiding is proposed in [35], in which the sensitive frequent itemset hiding is achieved by removing specific items. But, it generates the ghost rule, which may result in a higher loss of data utility.

Besides, the above mentioned methods adopt the traditional MOEA. However, with the increasing of data size, the performance of traditional MOEA often deteriorates rapidly. To address this large-scale optimization problem, a series of algorithms are presented [36]–[38]. The specific challenges of the large-scale optimization problem are examined empirically in [36], [37]. In addition, a self-evaluation evolution approach is proposed in [38]. In the future, the privacy-preserving minable data publication can be solved by these algorithms.

VI. Conclusion

This paper presents PMDP scheme to tackle the SARH problem to achieve privacy protection in minable data publication. The superiority of PMDP has been verified by intensive experimental results. PMDP provides more optional solutions of privacy protection for data publishers since it formulates the SARH problem as a MOP (two objects are data privacy and data utility). Besides, PMDP can provide a solution with a lower lost rule rate under the same hiding failure rate. In addition, PMDP has the advantage to be quick protection, and the proposed preprocessing mechanism allows PMDP to have a low computational cost.

REFERENCES

[1] L. Viktoratos, A. Tsadiras, and N. Bassiliades, "Combining community-based knowledge with association rule mining to alleviate the cold start problem in context-aware recommender systems," *Expert Systems with Applications*, vol. 101, pp. 78–90, 2018.

[2] L. Li, P. Ding, H. Chen, and X. Wu, "Frequent pattern mining in big social graphs," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–11, 2021.

[3] F. Kong, Y. Zhou, and G. Chen, "Traffic flow prediction method based on spatio-temporal feature mining," *Computer Science*, pp. 322–326, 2019.

[4] N. I. of Standards and Technology, "Advanced encryption standard," https://www.nist.gov/publications/advanced-encryption-standard-aes, 2001, published on November 26, 2001.

[5] N. S. Agency, "Data encryption standard," https://www.nsa.gov/News-Features/Declassified-Documents/Data-Encryption-Standard/, 1999, published on October 25, 1999.

[6] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. Mclachlan, A. Ng, B. Liu, and P. S. a. Yu, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008.

[7] S. L. Wang, B. Parikh, and A. Jafari, "Hiding informative association rule sets," *Expert Systems with Applications*, vol. 33, no. 2, pp. 316–323, 2007.

[8] F. N. Motlagh and H. Sajedi, "Mosar: A multi-objective strategy for hiding sensitive association rules using genetic algorithm," *Applied Artificial Intelligence*, pp. 823–843, 2016.

[9] B. Talebi and N. M. Dehkordi, "Sensitive association rules hiding using electromagnetic field optimization algorithm," *Expert Systems with Applications*, vol. 114, pp. 155–172, 2018.

[10] M. H. Afshari, M. N. Dehkordi, and M. Akbari, "Association rule hiding using cuckoo optimization algorithm," *Expert Systems with Applications*, vol. 64, pp. 340–351, 2016.

[11] H. Pang and B. Wang, "Privacy-preserving association rule mining using homomorphic encryption in a multikey environment," *IEEE Systems Journal*, vol. 15, no. 2, pp. 3131–3141, 2021.

[12] J. Wu, N. Mu, X. Lei, J. Le, and X. Liao, "Secedmo: Enabling efficient data mining with strong privacy protection in cloud computing," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1–1, 2019.

[13] A. Telikani, A. H. Gandomi, and A. Shahbahrami, "A survey of evolutionary computation for association rule mining," *Information Sciences*, vol. 524, pp. 318–352, 2020.

[14] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino, "Hiding association rules by using confidence and support," in *International Workshop on Information Hiding*, 2001, pp. 369–383.

[15] X. S. Yang, *Nature-Inspired Optimization Algorithms*. Elsevier Science Publishers B. V., 2014.

[16] Z.-H. Zhan, S.-H. Wu, and J. Zhang, "A new evolutionary computation framework for privacy-preserving optimization," *International Conference on Advanced Computational Intelligence*, pp. 220–226, 2021.

[17] G. M. Fan and H. J. Huang, "A novel binary differential evolution algorithm for a class of fuzzy-stochastic resource allocation problems," in *IEEE International Conference on Control and Automation*, 2017, pp. 548–553.

[18] F. Yang, N. Mu, X. Liao, and X. Lei, "Ea-hufim: Optimization for fuzzy-based high-utility itemsets mining," *International Journal of Fuzzy Systems*, pp. 1–17, 2021, doi: 10.1007/s40815-020-01003-8.

[19] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *ACM SIGMOD International Conference on Management of Data*, 2000.

[20] S. Li, N. Mu, J. Le, and X. Liao, "Privacy preserving frequent itemset mining: Maximizing data utility based on database reconstruction," *Computers & Security*, vol. 84, pp. 17–34, 2019.

[21] A. Telikani and A. Shahbahrami, "Data sanitization in association rule mining: An analytical review," *Expert Systems with Applications*, pp. 406–426, 2018.

[22] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, pp. 577–590, 2018.

[23] X. Liu, S. Wen, and W. Zuo, "Effective sanitization approaches to protect sensitive knowledge in high-utility itemset mining," *Applied Intelligence*, vol. 50, no. 1, pp. 169–191, 2020.

[24] P.-Q. Huang, Y. Wang, K. Wang, and K. Yang, "Differential evolution with a variable population size for deployment optimization in a uav-assisted iot data collection system," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 324–335, 2020.

[25] I. Fister, "Information cartography in association rule mining," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–17, 2021.

[26] S. Krishnamoorthy, G. S. Sadasivam, M. Rajalakshmi, K. Kowsalyaa, and M. Dhivya, "Privacy preserving fuzzy association rule mining in data clusters using particle swarm optimization," *International Journal of Intelligent Information Technologies*, vol. 13, no. 2, pp. 1–20, 2017.

[27] C. W. Lin, T. P. Hong, K. T. Yang, and S. L. Wang, "The ga-based algorithms for optimizing hiding sensitive itemsets through transaction deletion," *Applied Intelligence*, vol. 42, no. 2, pp. 210–230, 2015.

[28] A. Khan, M. S. Qureshi, and A. Hussain, "Improved genetic algorithm approach for sensitive association rules hiding," *World Applied Sciences Journal*, 2014.

[29] R. H. Qodmanan, M. Nasiri, and B. Minaei-Bidgoli, "Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence," *Expert Systems with Applications*, vol. 38, no. 1, pp. 288–298, 2011.

[30] Y. Chen, J. Zhong, L. Feng, and J. Zhang, "An adaptive archive-based evolutionary framework for many-task optimization," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 369–384, 2020.

[31] U. Ahmed, C.-W. J. Lin, G. Srivastava, R. Yasin, and Y. Djenouri, "An evolutionary model to mine high expected utility patterns from uncertain databases," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 19–28, 2021.

[32] M. A. Garcia-Vico, J. C. Carmona, P. Gonzalez, and J. d. M. Jess, "Moea-efep: Multi-objective evolutionary algorithm for extracting fuzzy emerging patterns," *IEEE Transactions on Fuzzy Systems*, pp. 2861–2872, 2018.

[33] L. Zhang, S. Yang, X. Wu, F. Cheng, Y. Xie, and Z. Lin, "An indexed set representation based multi-objective evolutionary approach for mining diversified top-k high utility patterns," *Engineering Applications of Artificial Intelligence*, pp. 9–20, 2019.

[34] V. E. Altay and B. Alatas, "Differential evolution and sine cosine algorithm based novel hybrid multi-objective approaches for numerical association rule mining," *Information Sciences*, pp. 198–221, 2021.

[35] Peng, Cheng, Lee, Ivan, Chun-Wei, Lin, Jeng-Shyang, and Pan., "Association rule hiding based on evolutionary multi-objective optimization." *Intelligent Data Analysis*, vol. 20, pp. 495–514, 2016.

[36] W. Hong, K. Tang, A. Zhou, H. Ishibuchi, and X. Yao, "A scalable indicator-based evolutionary algorithm for large-scale multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, pp. 525–537, 2018.

[37] W.-J. Hong, P. Yang, and K. Tang, "Evolutionary computation for large-scale multi-objective optimization: A decade of progresses," *International Journal of Automation and Computing*, pp. 1–15, 2021.

[38] P. Yang, K. Tang, and X. Yao, "Turning high-dimensional optimization into computationally expensive optimization," *IEEE Transactions on Evolutionary Computation*, pp. 143–156, 2018.

Fan Yang is currently pursuing the Ph.D. degree with the Department of Computer Science, Chongqing University, Chongqing, China. She received the B.S. and M.S. degrees from the College of Electronic and Information Engineering, Southwest University, Chongqing, China. Her research interests include data mining, evolutionary algorithm, and data trading.

Xinyu Lei (Member, IEEE) received the Ph.D. degree with the Department of Computer Science and Engineering, Michigan State University in 2021, the B.S. and M.S. degrees from the Department of Computing Science, Chongqing University, Chongqing, China. He worked in Texas A&M University at Qatar as a Research Assistant in 2013. In 2017, He worked as a Research Intern at Ford Motor Company. His current research focuses on machine learning and cybersecurity.

Junqing Le received the Ph.D. degree in intelligent computing and information processing and the M.S. degree in signal and information processing from Southwest University, Chongqing, China, in 2021 and 2017, respectively. He received the B.S. degree in software engineering from Southwest Jiaotong University, Chengdu, China, in 2014. From 2019 to 2021, he was a visiting scholar with George Mason University, Fairfax, VA, USA. His research interests include privacy protection, privacy machine learning, blockchain, and cloud computing security.

Nankun Mu (Member, IEEE) received the Ph.D. degree in computer science and technology from Chongqing University, Chongqing, China, in 2015, and the B.S. degree in software engineering and the M.S. degree in computer systems and structures from Chongqing University, Chongqing, China, in 2011 and 2013, respectively. His research areas include intelligent control, evolutionary computation, information security, and big data.

Xiaofeng Liao (Fellow, IEEE) received the Ph.D. degree in circuits and systems from the University of Electronic Science and Technology of China, Chengdu, in 1997, the B.S. and M.S. degrees in mathematics from Sichuan University, Chengdu, China, in 1986 and 1992, respectively. From 1999 to 2012, he was a Professor with Chongqing University, Chongqing, China. From January 2013 to November 2018, he was a Professor and the Dean of the College of Electronic and Information Engineering, Southwest University, Chongqing. He is currently a Professor and the Dean of the College of Computer Science, Chongqing University. He is also a Yangtze River Scholar of the Ministry of Education of China, Beijing, China. From November 1997 to April 1998, he was a Research Associate with the Chinese University of Hong Kong, Hong Kong. From October 1999 to October 2000, he was a Research Associate with the City University of Hong Kong, Hong Kong. From March 2001 to June 2001 and March 2002 to June 2002, he was a Senior Research Associate at the City University of Hong Kong. From March 2006 to April 2007, he was a Research Fellow at the City University of Hong Kong. He holds four patents, and published four books and over 300 international journal and conference papers. His current research interests include neural networks, nonlinear dynamical systems, bifurcation and chaos, and cryptography. Prof. Liao is an Associate Editor of IEEE Transactions on Cybernetics and IEEE Transactions on Neural Networks and Learning Systems.