BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 06/06 pp257–269
Volume 5, Number 3, September 2022
DOI: 10.26599/BDMA.2022.9020007

An Optimized Sanitization Approach for Minable Data Publication

Fan Yang and Xiaofeng Liao*

Abstract: Minable data publication is ubiquitous since it is beneficial to sharing/trading data among commercial companies and further facilitates the development of data-driven tasks. Unfortunately, the minable data publication is often implemented by publishers with limited privacy concerns such that the published dataset is minable by malicious entities. It prohibits minable data publication since the published data may contain sensitive information. Thus, it is urgently demanded to present some approaches and technologies for reducing the privacy leakage risks. To this end, in this paper, we propose an optimized sanitization approach for minable data publication (named as SA-MDP). SA-MDP supports association rules mining function while providing privacy protection for specific rules. In SA-MDP, we consider the trade-off between the data utility and the data privacy in the minable data publication problem. To address this problem, SA-MDP designs a customized particle swarm optimization (PSO) algorithm, where the optimization objective is determined by both the data utility and the data privacy. Specifically, we take advantage of PSO to produce new particles, which is achieved by random mutation or learning from the best particle. Hence, SA-MDP can avoid the solutions being trapped into local optima. Besides, we design a proper fitness function to guide the particles to run towards the optimal solution. Additionally, we present a preprocessing method before the evolution process of the customized PSO algorithm to improve the convergence rate. Finally, the proposed SA-MDP approach is performed and verified over several datasets. The experimental results have demonstrated the effectiveness and efficiency of SA-MDP.

Key words: data publication; data sanitization; association rules hiding; evolutionary algorithm

1 Introduction

Nowadays, data publication is ubiquitous because it can facilitate the development of data-driven services, such as classification[1], E-commerce recommendation[2], social network analysis[3,4], and so on. Noting that the published dataset may contain sensitive information, some commercial companies concerning the privacy leakage risks are reluctant to share their data for further mining tasks. To solve this problem, several privacy-preserving data publication methods are proposed[5]. Researchers protect the dataset by utilizing standard encryption algorithms (such as DES[6]) and then release it. Noting that the encrypted data is pseudo-random and hard to discover useful information. Intuitively, to support the minable data publication, a feasible approach is to weaken the encryption for improving the data utility while guaranteeing data privacy. Therefore, we aim to present a novel approach supporting the privacy-preserving minable data publication.

This paper considers a Top-3 data mining algorithm, i.e., association rule mining[7,8]. Now, let us briefly introduce this algorithm with the relevant privacy and security threats for a given transaction dataset that will be published. This dataset is collected from a market that has a series of items on sale. A data miner notices that the transactions contain both "Beer" and "Diapers" with a high frequency. In such a case, "Beer $\rightarrow$ Diapers" is an association rule being mined by the association rule mining algorithm. Based on this rule, the market

• Fan Yang and Xiaofeng Liao are with the College of Computer Science, Chongqing University, Chongqing 400044, China. E-mail: fany@cqu.edu.cn; xfliao@cqu.edu.cn.

* To whom correspondence should be addressed.
Manuscript received: 2022-03-10; accepted: 2022-03-20

manager can offer a discount package of buying "Beer" and "Diapers" together. This rule can be leaked to commercial competitors if it can be discovered from the published dataset. Thus, they may steal a lot of customers from the market once they pretend to provide the same discount strategy. Clearly, "Beer → Diapers" is sensitive and needs to be protected. To sum up, the target problem of this paper is how to provide non-sensitive association rules (NARs) mining while protecting sensitive association rules (SARs).

To solve the SARs protection (SARP) problem, several approaches have been proposed, but each of them has some limitations. First, they are low data privacy. To solve the SARP problem, some researchers protect the sensitive information by hiding the SARs, which is achieved by decreasing the frequency[9]. However, it is limited in terms of practicality because only one SAR can be protected at one time. Besides, some proposed approaches have a weak privacy protection[10]. Second, they are low data utility. To guarantee high data privacy, some researchers transform the SARP problem into a single objective optimization problem[11]. However, those proposed approaches just consider how to decrease the privacy leakage risks but ignore how to improve the data utility[12]. Namely, the data utility of those works is low. Third, they are low efficiency. To solve the SARP problem, some encryption-based approaches are also presented. One of these methods enables securely performing the data mining technique over the encrypted data[13], and another can outsource the data mining tasks to a third-party[14]. However, the homomorphic encryption-based approaches are time-consuming, which is also limited in implementation on a large dataset.

In this paper, we design SA-MDP, an optimized sanitization approach for minable data publication. SA-MDP enables mining the NARs while protecting the SARs. To make an acceptable trade-off between the data utility and the data privacy, SA-MDP considers the SARP from a perspective of the multi-objective optimization problem (MOP). How to maximize the data utility and how to minimize the data privacy leakage risks are two important objectives. Based on this, SA-MDP designs a customized particle swarm optimization (PSO) algorithm, in which the particle can learn from the best solution. Specifically, SA-MDP presents a preprocessing mechanism to remove the unnecessary transactions. Besides, SA-MDP introduces the concept of particle splitting, which enables a particle to produce several child particles. Furthermore, SA-MDP designs a weighted-fitness function to quantify the quality of particles.

To address the SARP problem, SA-MDP has two major challenges to be addressed. First, it is challenging to avoid the solution being trapped into local optima. Specifically, PSO may cause premature convergence while solving a complex problem. The reason lies that it is prone to be trapped into local optima. Considering the evolution method greatly affects the search direction[15], we propose a proper design to solve this challenge. SA-MDP introduces the concept of particle splitting, which enables a particle to produce several child particles to avoid being trapped into local optimal. Second, it is also challenging to improve the convergence rate. For any design of the multi-objective evolutionary algorithm (MOEA), a significant problem is how to guarantee the convergence rate to find the optimum solution. Meanwhile, there are many sparse published datasets, which is time-consuming to process. To tackle this challenge, PSO enables the particle to learn from the best solution, which guarantees the convergence rate. Besides, the proposed preprocessing method can decrease the computational costs by filtering out the irreverent transactions. Thus, the convergence rate can be further improved.

Our contributions are summarized as follows.

(1) To address the SARP problem, we present SA-MDP. SA-MDP considers the SARP problem from a perspective of MOP to make an acceptable trade-off between the data utility and the data privacy. Accordingly, we design a customized PSO algorithm to efficiently solve the MOP.

(2) To avoid the solution being trapped into a local optimum, SA-MDP presents a proper design of the evolution method. SA-MDP introduces the concept of particle splitting, which enables a particle to produce several child particles. Thus, the exploration ability of SA-MDP can be improved.

(3) To improve the convergence rate, SA-MDP designs a novel preprocessing mechanism to remove the unnecessary transactions. Moreover, PSO enables particle to learn from the best solution, which improves the exploitation ability.

The rest of this paper is organized as follows. In Section 2, we introduce some preliminaries and give a formal problem statement. In Section 3, we describe the SA-MDP approach. In Section 4, we analyze the experimental results. In Section 5, we review the related works. In Section 6, we conclude this paper.

2 Preliminaries and Problem Statement

In this section, we introduce association rule mining and PSO algorithm with several necessary definitions. Then, the formal problem statement is given. Some frequently used notations in this paper are summarized in Table 1.

2.1 Association rules mining

The dataset that will be published contains the data records collected from n individuals, and each data record consists several items. Hereinafter, we let $\mathcal{T} = \{t_1, \ldots, t_n\}$ denote the dataset and $\mathcal{S}$ represent all the items belonging to $\mathcal{T}$. Besides, the ith data record is labeled by t_i, and the jth item is labeled by s_j.

Suppose that X and Y are two subsets of $\mathcal{S}$, they have $X \cap Y = \varnothing$. Generally, an association rule can be represented by $(X \rightarrow Y)$, where X and Y are the left-hand side (LHS) and the right-hand side (RHS) of this rule, respectively. To measure the usefulness of this rule, there are two definitions will be used, i.e., Support and Confidence.

Definition 1. (Support). The Support of itemset XY is

$$Sup_{(XY)} = \frac{|X \cup Y|}{|\mathcal{T}|} \times 100\% \tag{1}$$

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

Definition 2. (Confidence). The Confidence of rule $(X \rightarrow Y)$ is

$$Conf_{(X \rightarrow Y)} = \frac{|X \cup Y|}{|X|} \times 100\% = \frac{Sup_{(XY)}}{Sup_{(X)}} \times 100\% \tag{2}$$

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

Table 1 A summary of the frequently used notations.

Notation	Meaning
$\mathcal{T}$	Dataset, the i-th transaction is t_i
$\mathcal{S}$	Itemset, the j-th item is s_j
$minS, minC$	Thresholds of support and confidence
$R_s, \overline{R_s}$	Sensitive and non-SARs
C_T, C_R	Critical transactions and critical rules
S_I	Sensitive items
p_i^n	i-th solution at the n-th generation
$\mathcal{P}^n = \{p_i^n\}$	Population at the n-th iteration
$\mathcal{F}(p_i^n)$	Fitness value of p_i^n
$\mathcal{N}_{pop}$	Naximum population size
$\mathcal{N}_{gen}$	Number of generations
$\mathcal{T}_{fit}$	Maximum times of fitness can not be updated
$\mathcal{T}_{gen}$	Maximum number of generations

Technically, the association rules mining is performed as follows. Given a rule $(X \rightarrow Y)$, let Support measure the frequency that both X and Y occur in the dataset $\mathcal{T}$, and let Confidence measure the frequency that Y occurs when X occurs. According to the above definitions, we can define the association rule.

Definition 3. (Association rule). An association rule $(X \rightarrow Y)$ holds if

$$Sup_{(XY)} \geqslant minS \text{ and } Conf_{(X \rightarrow Y)} \geqslant minC \tag{3}$$

The miner can adjust $minS$ and $minC$ to mine useful rules according to their demands. We illustrate the association rules mining with an example dataset $\mathcal{T}^*$, which is given in Table 2.

Example. The dataset $\mathcal{T}^*$ contains 10 transactions, and the itemset $\mathcal{S}^* = \{A, B, C, D, E, F\}$ and $|\mathcal{S}^*| = 6$. Let the minimum support threshold $minS$ and the minimum confidence threshold $minC$ be 30% and 70%, respectively. The mined 11 association rules are shown in Table 3.

2.2 Particle swarm optimization

This paper adopts the PSO algorithm to solve the SARP problem. For any problem, the canonical PSO algorithm formulates each candidate solution as a particle and guides all particles to fly toward the best-so-far position g_{best}. Formally, suppose there are N particles, the i-th particle is determined by two vectors, i.e., the position vector p_i^n and the velocity vector v_i^n, where $i \in \{1, 2, \ldots, N\}$, and n denotes the current generation number.

Generally, PSO initilizes the position and the velocity of each particle by a uniformly random manner[16]. Specifically, for the next generation, the new positions

Table 2 The example dataset $\mathcal{T}^*$.

TID	Items	TID	Items
t_1	B, C, D, F	t_6	A, D, E
t_2	A, B, E	t_7	C, D, E, F
t_3	A, C	t_8	A, C, F
t_4	B, D, F	t_9	B, D, F
t_5	A, B, C, D, E, F	t_{10}	E, F

Table 3 The mined rules of dataset $\mathcal{T}^*$.

(%)

Rules	Support	Confidence	Rules	Support	Confidence
$B, D \rightarrow F$	40	100	$C \rightarrow F$	40	80
$B, F \rightarrow D$	40	100	$B \rightarrow D, F$	40	80
$C, D \rightarrow F$	30	100	$D, F \rightarrow B$	40	80
$D \rightarrow F$	50	83.33	$C, F \rightarrow D$	30	75
$B \rightarrow D$	40	80	$F \rightarrow D$	50	71.43
$B \rightarrow F$	40	80	/	/	/

and velocities of each particles can be updated by the following two equations.

$$v_i^{n+1} = c_1 r_1 (p_{best} - p_i^n) + c_2 r_2 (g_{best} - p_i^n) + c_3 v_i^n \tag{4}$$

$$p_i^{n+1} = p_i^n + v^{n+1,i} \tag{5}$$

where c_1, c_2, and c_3 are three positive numbers used to adjust the updating direction of velocity. r_1 and r_2 are two random numbers. Besides, p_{best} and g_{best} are the best-previous position and the best-so-far position, respectively. The current position p_i^n will be updated once particles find a better one. But it is worth noting that if particles do not find a position better than g_{best}, all particles will search in this small solution space, i.e., solutions are trapped into a local optimum[16].

2.3 Problem statement

This paper focuses on the SARP problem and aims at presenting a novel approach that can protect SARs while keeping a high data utility. Motivated by this, SA-MDP models SARP problem as an MOP problem (the two objectives are the data privacy and the data utility, respectively). SA-MDP is performed based on the following conditions. (i) the data publisher has already discovered all association rules that have existed in the given dataset; (ii) the data publisher has divided the association rules into SARs and NARs. To sum up, SA-MDP sanitizes the given dataset for hiding SARs while maintaining NARs under the same Support and Confidence threshold, and the SARP problem can be defined as follows.

Input:

(1) $|\mathcal{T}|$: the original dataset,

(2) R_s: the sensitive association rules,

(3) $minS, minC$: the minimum thresholds of Support and Confidence.

Output:

(1) g_{best}: the optimal hiding scheme,

(2) $\mathcal{F}(g_{best})$: the fitness of the optimal solution.

To address this problem, SA-MDP is presented as follows.

3 SA-MDP Design

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

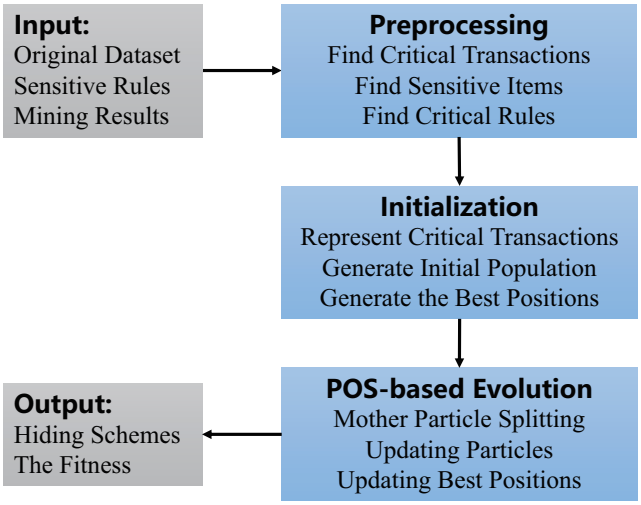


Fig. 1 The flowchart of SA-MDP.

3.1 Preprocessing

In this section, we introduce the preprocessing strategy. Technically, the data publisher can protect an SAR by reducing its Support or Confidence to drop below the related mining thresholds[5]. To this end, the data publisher can delete/add some items from/into the original dataset. Unfortunately, the mentioned methods cause the following three side-effects.

- Hiding-Failure. Some SARs still can be discovered from the original and the sanitized datasets.
- Lost-Rule. Some NARs can be discovered from the original dataset, but they are lost in the sanitized datasets.
- Fake-Rule. Some fake association rules (FARs) cannot be mined from the original dataset, but they are generated in the sanitized datasets.

These side-effects can be measured by the following three metrics.

- Hiding-Failure Rate (HFR): $\alpha = \dfrac{|R_s \cap R_s'|}{|R_s|}$,
- Lost-Rule Rate (LRR): $\beta = \dfrac{|\overline{R_s} \cap \overline{R_s'}|}{|\overline{R_s}|}$,
- Fake-Rule Rate (FRR): $\gamma = \dfrac{|\complement_{\overline{R_s'}}\{\overline{R_s} \cap \overline{R_s'}\}|}{|\overline{R_s}|}$,

where R_s and $\overline{R_s}$ are the sets of SARs and NARs, respectively. Meanwhile, R_s' and $\overline{R_s'}$ are the sets of SARs and NARs that obtained after sanitization process, respectively. In addition, $|\cdot|$ denotes the number of elements in the related set, and $\complement_A B$ means the complement of the set B relative to set A.

It is worth pointing out that the Support of any FARs will not be increased if we remove the items[5].

Algorithm 1: SA-MAP

Input: $\mathcal{T}$, R_s, $minS$, and $minC$
Output: g_{best} and $\mathcal{F}(g_{best})$

Generate the critical transactions C_T, the sensitive items S_I, and the critical rules C_R; // Preprocessing
Calculate the upper and the lower bound of the number of transactions can be modified, i.e., $\mathcal{M}_{\max}$ and $\mathcal{M}_{\min}$;
// Initialization
Generate the initial population $\mathcal{P}^0$ and calculate the relevant fitness $\mathcal{F}(\mathcal{P}^0)$;
Generate the best-previous position p_{best} and the best-so-far position g_{best};
repeat
  // Evolution
  for $i = 1, 2, \ldots, \mathcal{N}_{pop}$ **do**
    Calculate the number of child particles $s_i = \lfloor (\mathcal{S}_{\max} - \mathcal{S}_{\min}) \times r_1 \rfloor - \mathcal{S}_{\min}$;
  for $i = 1, 2, \ldots, \mathcal{N}_{pop}$ **do**
    Update the velocity $v^{n+1,i} \leftarrow [p_{best} - p_i^n] \cup [g_{best} - p_i^n]$;
    Update the position $p^{n+1,i} \leftarrow \{p_i^n, \text{null}\} \times r_2 \cup v^{n+1,i}$;
    Calculate the distance of the child particle which can be achieved $m_i = \left\lfloor \phi \times \frac{s_i}{\sum_{i=1}^{\mathcal{N}_{pop}} s_i} \right\rfloor \times (\mathcal{M}_{\max} - \mathcal{M}_{\min})$;
    for $j = 2, 3, \ldots, s_i$ **do**
      Randomly selecting m_i elements from the C_T and changing them ;
  Calculate the fitness $\mathcal{F}(\mathcal{P}^{*,i})$ and generate the new best-previous position p_{best} ;
  if $\mathcal{F}(p_{best}) \leqslant \mathcal{F}(g_{best})$ **then**
    Update the best-so-far position $g_{best} \leftarrow p_{best}$ // Updating
  else
    $\mathcal{N}_{fit} \leftarrow \mathcal{N}_{fit} + 1$;
  $\mathcal{N}_{gen} \leftarrow \mathcal{N}_{gen} + 1$;
until $\mathcal{T}_{fit} \leqslant \mathcal{N}_{fit}$ ***or*** $\mathcal{T}_{gen} \leqslant \mathcal{N}_{ite}$;
// Termination

Hence, we prefer to hide SARs by removing items than randomly modifying the dataset. In addition, it will cause more serious side-effects if the data publisher changes the LHS of SARs other than the RHS of SARs. Thus, SA-MDP sanitizes the dataset by deleting some items in the RHS of SARs. Next, we give three definitions for preprocessing.

Definition 4. (Critical transaction). A transaction is a critical transaction (C_T) if it supports one or more SARs.

Specifically, "t_i supports $X \rightarrow Y$" means t_i contains all the items of $X \cup Y$. Then, by Definition 4, critical transactions are composed of those SAR-related transactions, and the rest transactions are filtered out, so the solution space can be reduced.

Definition 5. (Sensitive item). An item is a sensitive item (S_I) if it is the only item in the RHS of SAR or it has the highest frequency in the RHS of SAR and the lowest frequency in NAR.

SA-MDP tends to remove the S_I to hide SARs. That is to say, only the S_I-related NARs will be affected. Therefore, we present the third definition.

Definition 6. (Critical rule). A critical rule (C_R) is the NAR which contains any S_I.

According to Definition 6, SA-MDP can evaluate the LRR by evaluating the loss of C_R. To sum up, the preprocessing can be summarized in three steps.

(1) SA-MDP removes useless information and leaves C_T.

(2) SA-MDP computes the frequency of the RHS of SARs and finds S_I.

(3) SA-MDP finds all NARs containing S_I and labels them as C_R.

After the preprocessing, SA-MDP can reduce the search space to improve the convergence rate.

Example. We continue to illustrate the preprocessing, step by step. We select the following two rules as SARs that need to be protected.

$$(1)\ B \rightarrow F \quad \text{and} \quad (2)\ C \rightarrow F$$

Then, the preprocessing is performed as follows.

First, based on Definition 4, SA-MDP removes 4 transactions, i.e., t_2, t_3, t_6, t_{10}. The filtered out transactions are given in Table 4. Then, according to Definition 5, SA-MDP determines the item F is the S_I

Table 4 The critical transactions $\mathcal{C}^*_{\mathcal{T}}$.

TID	Items	TID	Items
t_1	B, C, D, F	t_7	C, D, E, F
t_4	B, D, F	t_8	A, C, F
t_5	A, B, C, D, E, F	t_9	B, D, F

for the two SARs. Next, according to Definition 6, NARs containing F are identified as the C_R. Thus, we have 8 C_R, which are shown in Table 5.

The data publisher can obtain that $\frac{|\mathcal{C}^*_{\mathcal{T}}|}{|\mathcal{T}|} = 60\%$ and $\frac{|\mathcal{C}^*_{\mathcal{R}}|}{|\overline{R_S}|} = 73\%$. In other words, the presented preprocessing can reduce the amount of data and further reduce the search space.

3.2 Fitness function

In this section, we show how to design the fitness function based on the preprocessing strategy. Specifically, we design the fitness function based on the above-mentioned three side-effects, in which α denotes HFR and β denotes LRR. Especially, α increases by 1 when one SAR in R_s is failed to be protected and β increases by 1 when one NAR in $\overline{R_s}$ is lost after the hiding process. They are two coarse-grained evaluation metrics since α and β are changed by modifying multiple transactions. Therefore, to provide a fine-grained evaluation for the solution quality, we present a new evaluation metric, optimal sanitation distance (OSD) δ, which is calculated as follows:

$$\delta = \frac{\sum_{l=1}^{R_s} a_l}{|R_s|} + \frac{\sum_{u=1}^{\overline{R_s}} b_u}{|\overline{R_s}|} \tag{6}$$

where a_l is calculated as

$$a_l = \max\{\min\{Conf_l - minC, Sup_l - minS\}, 0\} \tag{7}$$

a_l can reflect how much at least the Confidence or the Support needs to be decreased for hiding the l-th SAR. Besides, b_u is calculated as

$$b_u = \max\{\max\{minC - Conf_u, minS - Sup_u\}, 0\} \tag{8}$$

b_u shows how much at most the Confidence or the Support needs to be increased for avoiding the loss of the u-th NAR. Recall that SA-MDP hides SAR by removing its RHS, no FAR will be generated. Thus, the FRR of SA-MDP identically equals 0. Considering three side-effects, the fitness function can be represented as

$$\min\ \mathcal{F}(p) = w_1 \times \alpha + w_2 \times \beta + w_3 \times \delta \tag{9}$$

Especially, in this paper, we set $w_1 = w_2 = w_3 = 1$. The data publisher can adjust the weights according to the practical demands. The fitness function $\mathcal{F}(p)$ provides a precise quantify of the solutions' quality.

Table 5 The critical rules $\mathcal{C}^*_{\mathcal{R}}$.

(%)

Rules	Support	Confidence	Rules	Support	Confidence
$B, D \rightarrow F$	40	100.00	$B \rightarrow D, F$	40	80.00
$B, F \rightarrow D$	40	100.00	$D, F \rightarrow B$	40	80.00
$C, D \rightarrow F$	30	100.00	$C, F \rightarrow D$	30	75.00
$D \rightarrow F$	50	83.33	$F \rightarrow D$	50	71.43

3.3 Initialization

In this section, we introduce the initialization method, which can be divided into the following three steps.

(1) Represent the critical transaction with a binary matrix $\mathcal{C}_{\mathcal{T}}$. In the preprocessing, the data publisher has filtered out the critical transaction from $\mathcal{T}$ according to Definition 4. Let $\mathcal{C}_{\mathcal{T}}$ denote the set of the critical transaction, which will be transformed into a $|\mathcal{C}_{\mathcal{T}}| \times |\mathcal{S}|$ binary matrix. In such matrix, the (i, j)-th element equals 1, which means that the item s_j exists in transaction t_i.

(2) Generate an initial population $\mathcal{P}^0$. According to $\mathcal{C}_{\mathcal{T}}$, the data publisher generates an initial particle $p^0_1 = \{1, 2, \ldots, |\mathcal{C}_{\mathcal{T}}|\}$. Then, to generate the initial population $\mathcal{P}^0 = \{p^0_1, \ldots, p^0_{\mathcal{N}_{pop}}\}$, the data publisher randomly selects the elements in p^0_1, and those selected elements compromise a new particle. This process will perform $\mathcal{N}_{pop} - 1$ times, where $\mathcal{N}_{pop}$ is the population size and can be empirically set. $\mathcal{P}^0$ are constituted by the $\mathcal{N}_{pop}$ particles.

(3) Generate the initial best-previous position p_{best} and the initial best-so-far position g_{best}. SA-MDP calculates the fitness of each particle in $\mathcal{P}^0$. The particle having the smallest fitness is the initial best previous position p_{best}, which also is the best-so-far position g_{best}.

Before the evolution is run, SA-MDP can further reduce the dimension of solution space by controlling the modifying numbers. Specifically, according to the Support and Confidence of SARs, SA-MDP can determine the upper and the lower bound of the number of transactions as $\mathcal{M}_{\max}$ and $\mathcal{M}_{\min}$, respectively. Let d_1 denote the maximum Support which needs to be reduced, and d_2 denotes the maximum Confidence which needs to be reduced. Suppose $X \rightarrow Y$ is the SAR R_S that needs to be protected, we have to remove the Y-related items to hold that

$$Sup_{R_l} \times |T| - d_1 < minS \times |T| \tag{10}$$

or

$$Conf_{R_l} \times Sup_{R_{l_X}} |T| - d_2 < minC \times Sup_{R_{l_X}} |T| \tag{11}$$

To sum up, d_1 and d_2 can be calculated as follows.

$$d_1 = \max_{R_l \in R_s} (\lceil (Sup_{R_l} - minS) \times |T| \rceil) \quad (12)$$

$$d_2 = \max_{R_l \in R_s} (\lceil (Conf_{R_l} - minC) \times Sup_{R_{l_X}} |T| \rceil) \quad (13)$$

where $\lceil \cdot \rceil$ is the ceiling function.

Then, $\mathcal{M}_{\min}$ is obtained as

$$\mathcal{M}_{\min} = \min(d_1, d_2) \quad (14)$$

Based on $\mathcal{M}_{\min}$, SA-MDP decreases the number of the modified transactions such that the data utility can be guaranteed to a certain degree. In SA-MDP, $\mathcal{M}_{\max}$ is set as the number of C_T, i.e., $\mathcal{M}_{\max} = |\mathcal{C}_\mathcal{T}|$.

Example. According to the above discussion, the initialization is achieved in the following three steps. The first step is to represent the obtained $\mathcal{C}^*_\mathcal{T}$ with a binary matrix. Note that t_1 contains 4 items (i.e., B, C, D, and F), so t_1 can be represented by 011101. Besides, t_4, t_5, t_7, t_8, and t_9 are also denoted by a binary vector, respectively. The transformed binary matrix is obtained and is shown in Table 6. The second step is to generate an initial population $\mathcal{P}^0 = \{p^0_1, \ldots, p^0_{\mathcal{N}_{pop}}\}$. From the obtained matrix, we pick the S_I-related columns as the data that needs to be removed. Let the transaction identifier of the removed items as the position of the particle. Thus, we can have an initial particle $p^0_1 = \{\text{null}\}$, which means not to remove any items. Then, we randomly select the transactions in $\mathcal{C}^*_\mathcal{T}$ to sanitize. For instance, t_4 and t_9 are selected. Then, the identifier of those transactions constitutes a new position of particle $p^0_2 = \{4, 9\}$. We perform $\mathcal{N}_{pop} - 1$ times and then obtain the initial population. The third step is to calculate the fitness of these initial particles. The particle having the smallest fitness is the initial best previous position p_{best}, which also is the best-so-far position g_{best}.

3.4 Evolution

This section briefly introduces the evolution process of SA-MDP. The evolution includes: mother particle splitting, updating the velocity and the position, and updating the best positions. Suppose the current particle that needs to be updated is p^n_i, the main steps are described as follows.

Table 6 Represent $\mathcal{C}^*_\mathcal{T}$ with a binary matrix.

TID	A	B	C	D	E	F
t_1	0	1	1	1	0	1
t_4	0	1	0	1	0	1
t_5	1	1	1	1	1	1
t_7	0	0	1	1	1	1
t_8	1	0	1	0	0	1
t_9	0	1	0	1	0	1

(1) Mother particle p^n_i splitting. Suppose each mother particle p^n_i can be split into several child particles p^{n+1}_i owning to the same property with p^n_i. Especially, let $\mathcal{S}_{\max}$ denote the maximum number of the child particles and $\mathcal{S}_{\min}$ denote the minimum number of the child particles. Thus, the number of the child particles of each mother particle can be computed as follows.

$$s_i = \lfloor (\mathcal{S}_{\max} - \mathcal{S}_{\min}) \times r_1 \rfloor - \mathcal{S}_{\min} \quad (15)$$

where $\mathcal{S}_{\max}$ and $\mathcal{S}_{\min}$ are set by the data publisher, $\lfloor \cdot \rfloor$ is the floor function, and r_1 is a random number.

(2) Updating the velocity and the position. SA-MDP adopts two updating methods: (a) the directional learning and (b) the random splitting.

(a) The directional learning method. For the particle p^n_i, $i = 1, 2, \ldots, s_i$, the specific particle updating process is performed as Eqs. (16) and (17), respectively.

$$v^{n+1}_i = [p_{best} - p^n_i] \cup [g_{best} - p^n_i] \quad (16)$$

$$p^{n+1}_i = \{p^n_i, \text{null}\} \times r_2 \cup v^{n+1}_i \quad (17)$$

The velocity of particles is updated as Eq. (16), where p_{best} is the best-previous position and g_{best} is the best-so-far solution. Accordingly, the position of particles is updated as Eq. (17), where null is an empty dimension and means no transactions will be modified, and r_2 is a randomly generated number.

(b) The random splitting method. For the particle p^n_i, $i = 1, 2, \ldots, s_i$, during the specific particle updating process, SA-MDP first calculates the modifying distance as Eq. (18).

$$m_i = \left\lfloor \phi \times \frac{s_i}{\sum_{i=1}^{\mathcal{N}_{pop}} s_i} \right\rfloor \times (\mathcal{M}_{\max} - \mathcal{M}_{\min}) \quad (18)$$

where ϕ is the total number of child particles that can be generated at most. Then, SA-MDP selects m_i elements from the C_T and changes them randomly.

(3) Updating the best positions (p_{best} and g_{best}). SA-MDP calculates the fitness $\mathcal{F}(\mathcal{P}^*_i)$ and generates the new best-previous position p_{best}. Then, if $\mathcal{F}(p_{best}) \leqslant \mathcal{F}(g_{best})$, SA-MDP updates the best-so-far solution g_{best} by p_{best}. Otherwise, there is no updating during this generation, then the value of $\mathcal{N}_{fit}$ is incremented by 1.

If the mother particle can be split into multiple child particles, it will improve the diversity of the solution and the exploration ability of the algorithm. Therefore, there are more opportunities to escape from the local optimal solution and find the optimal particle. However, any particles certainly cannot split infinitely. Hence, SA-MDP has $\mathcal{S}_{\max}$ and $\mathcal{S}_{\min}$, which denote the maximum

and minimum number that a particle can be split into, respectively. The number of child particles s_i split from the mother particle is randomly generated as Eq. (15).

In Step 2, the velocity and the position are updated. Specifically, the SA-MDP approach proposed in this paper adopts two updating methods: one is the directional learning of particles, which can improve the exploitation ability of the algorithm; the other is random splitting of particles, which can improve the exploration ability of the algorithm. Besides, in order to adapt to the evolutionary environment, the position of the particles will be randomly deflected during the splitting process. The number of deflection dimensions of the particle position will be determined according to $\mathcal{M}_{\max}$ and $\mathcal{M}_{\min}$.

Example. Suppose $p_{best} = \{1, 4, 9\}$, $g_{best} = \{4, 7, 9\}$ and, $p_i^n = \{4, 9\}$. First, we let 5 and 2 be the values of $\mathcal{S}_{\max}$ and $\mathcal{S}_{\min}$, respectively. Second, we update particle. By the directional learning method, the result of $p_{best} - p_i^n$ is $\{1\}$ and the result of $g_{best} - p_i^n$ is $\{7\}$. Then, $v^{n+1,i} = \{1, 7\}$. On the other hand, a target set $\{p_i^n, \text{null}\} = \{4, 9, \text{null}\}$. If $r_2 = 0.4 \times 3 = 1.2$, we randomly select one dimension from the target set, i.e., $\{4\}$. Then, we can obtain the updated position $p^{n+1,i} = \{4\} \cup \{1, 7\} = \{1, 4, 7\}$. Moreover, by the randomly splitting method, SA-MDP calculates the number of dimensions that need to be changed for each child particle $\mathcal{M}_{\max}$ and $\mathcal{M}_{\min}$. The data publisher can set $\mathcal{S}_{\max}$ and $\mathcal{S}_{\min}$. Suppose m_i is 2 for p_i^n, SA-MDP can obtain $p_i^{n+1} = \{4, \text{null}\}$. Third, after the user obtains new child particles, SA-MDP calculates the relevant fitness and updates the best positions p_{best} and g_{best}.

3.5 Termination

The termination is determined by two parameters, i.e., $\mathcal{T}_{fit}$ and $\mathcal{T}_{gen}$. Namely, if $\mathcal{N}_{fit} \geqslant \mathcal{T}_{fit}$ or $\mathcal{N}_{gen} \geqslant \mathcal{T}_{gen}$, SA-MDP terminates. Specifically, $\mathcal{N}_{fit}$ counts the number of times the best-so-far solution g_{best} has not been updated and $\mathcal{N}_{gen}$ denotes the number of generations. SA-MDP will repeat until at least one of these two termination conditions is satisfied.

4 Performance Evaluation

In this section, we evaluate the performance of SA-MDP on the data utility and the data privacy, i.e., the HFR and LRR. To this end, substantial experiments are conducted to evaluate the effectiveness and efficiency of the proposed SA-MDP approach. The experimental results are compared with COA4ARH[12]. Next, let us introduce some experimental-related settings.

- **Implementation.** SA-MDP in the experiments is implemented in 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.** SA-MDP is conducted on three typical public datasets[17], i.e., Chess, Mushroom, and BMS1. Table 7 shows the characteristics of these three datasets. To evaluate the performance of the SA-MDP, the $minS$ and $minC$ are set differently for each dataset because we need to ensure the number of SARs is appropriate.
- **Parameters setting.** In the experiments, we set the number of generations $\mathcal{N}_{gen}$ to 15 and the particle population size $\mathcal{N}_{pop}$ is set to 100. For the fitness function, we set $w_1 = w_2 = w_3 = 1$. To determine which S_I needs to be removed for protecting the SARs, we select the particle that has the smallest fitness to delete items.
- **Metrics.** To evaluate the performance of SA-MDP, we consider two side-effects as the measured metrics, i.e., (1) hiding failure rate (HFR) and (2) lost rule rate (LRR). Besides, to show the efficiency, we utilize the third metric, i.e., (3) running time.

4.1 Trade-off between data privacy and data utility

In this section, we conduct a comparison on the performance of SA-MDP and COA4ARH in terms of the trade-off between the data privacy and the data utility. Recall that the HFR and the LRR are the two measured metrics, Fig. 2 shows the evolution process of the HFR and the LRR obtained by SA-MDP and COA4ARH. Next, a detailed analysis is given.

For the three public datasets, Fig. 2 shows the evolution process of the HFR and the LRR obtained by SA-MDP and COA4ARH. We observe that the HFR of the COA4ARH algorithm decreases monotonically and eventually converges to 0 while the LRR of the COA4ARH algorithm decreases steadily. Correspondingly, in the proposed SA-MDP approach, both HFR and LRR show large changes. But it is worth mentioning that for the proposed SA-MDP approach, the sum of HFR and LRR shows a monotonically decreasing trend. In Fig. 2, as can be seen, the proposed SA-MDP

Table 7 Characteristics of datasets.

Dataset	Trans.	Items	Rules	$minS$ (%)	$minC$ (%)	Type
Chess	3196	74	10 742	90	90	Dense
Mushroom	8416	119	3828	40	70	Dense
BMS1	59 601	497	17 678	0.1	5	Sparse

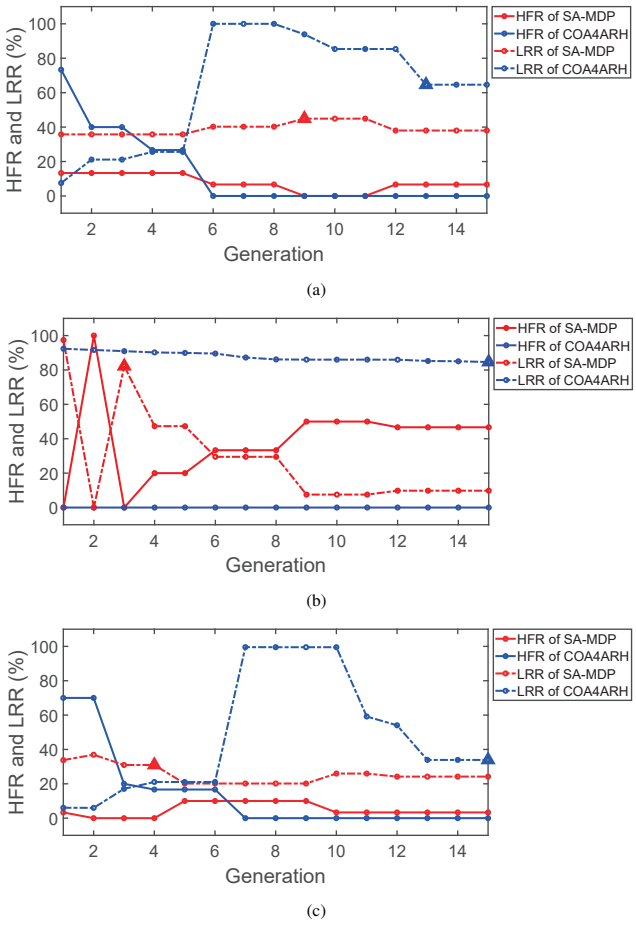


Fig. 2 Evolution processes of SA-MDP and COA4ARH on (a) Chess, (b) Mushroom, and (c) BMS1 datasets.

approach can completely protect all SARs in the three public datasets.

Take the results obtained over the Mushroom dataset as an example, we consider the best solution in terms of HFR, the best solution of the SA-MDP approach is obtained at the 3rd generation and has LRR $\beta = 82.18\%$ and HFR $\alpha = 0\%$ while the best solution of the COA4ARH algorithm is obtained at the 15th generation and has LRR $\beta = 84.62\%$ and HFR $\alpha = 0\%$. Note that, at the 15th generation, the HFR of SA-MDP is larger than 0 while that of COA4ARH equals 0, the reason lies that the best solution obtained by SA-MDP is determined by the weighted-fitness. In other words, SA-MDP enables the data publisher to decide which side-effect is more important, but COA4ARH only provides a privacy-first solution.

To sum up, SA-MDP performs better than other algorithms in terms of personalization. Namely, SA-MDP has better performance on balancing the trade-off between the data privacy and the data utility than COA4ARH.

4.2 Performance on data privacy

In this section, we demonstrate the performance of the proposed SA-MDP approach on the data privacy over three different datasets. Recall that we use the HFR as the measured metric of the data privacy, we conduct a comparison with COA4ARH in terms of HFR, and we choose the HFR of the solution which has the lowest fitness value. The experimental results are shown in Fig. 3. Next, a detailed analysis is given.

Figure 3 shows that the SA-MDP approach can achieve the same protective effect as the COA4ARH algorithm in terms of the HFR. Specifically, for the dense or sparse dataset, the SA-MDP approach can successfully protect all SARs as shown in Fig. 3. In addition, according to Fig. 2, the presented SA-MDP approach allows a certain hidden failure rate to further reduce the loss rate. This is impossible for the COA4ARH algorithm.

4.3 Performance on data utility

In this section, we demonstrate the performance of SA-MDP on the data utility. Recall that we use LRR as the measured metric, the results of SA-MDP in terms of LRR are compared with that of COA4ARH, and we choose the LRR of the solution that has the lowest fitness value at each generation. The experimental results are shown in Fig. 4. The detailed analysis is given as follows.

Figure 4 demonstrates that the SA-MDP approach performs better than the COA4ARH algorithm in terms of LRR, since the SA-MDP approach loses less NARs than the COA4ARH algorithm. Specifically, the LRRs

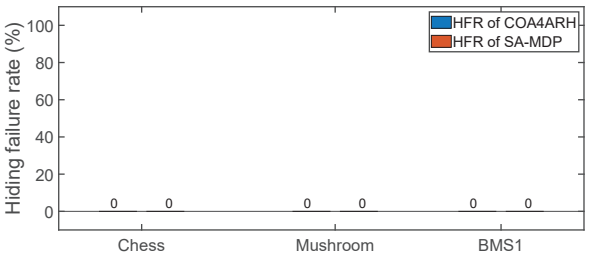


Fig. 3 HFR comparison of SA-MDP and COA4ARH.

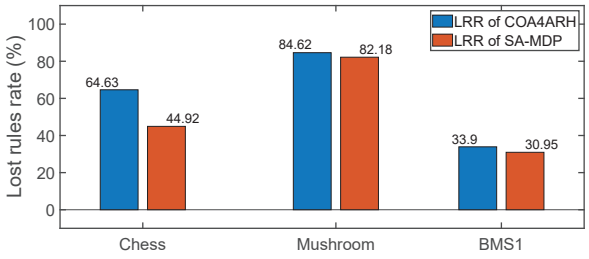


Fig. 4 LRR comparison of SA-MDP and COA4ARH.

obtained by the COA4ARH algorithm over the three datasets are 64.63%, 84.62%, and 33.90%, respectively. Meanwhile, the LRRs obtained by the proposed SA-MDP approach over the three datasets are 44.92%, 82.18%, and 30.95%, respectively.

Considering the trade-off between the LRR and the HFR, more SARs are protected, while more related association rules may be lost. Therefore, during the sanitization process achieved by deleting the RHS of SARs, there are more NARs that may be lost. According to Fig. 2, we know that this trade-off is solved by the SA-MDP approach and the COA4ARH algorithm. However, in the COA4ARH algorithm, the users just ignore the effect of LRR of the SA-MDP algorithm. The strategy leaves the users no choice to show the private preference of the data utility and the data privacy. To address this problem, the proposed SA-MDP approach introduces the weighted-fitness to handle the trade-off between the data utility and the data privacy. Besides, as we can know, the COA4ARH algorithm loses fewer NARs than GA-based algorithms. In other words, the SA-MDP approach performs better than other algorithms in terms of LRR. To sum up, the proposed SA-MDP approach performs better than the COA4ARH algorithm in terms of LRR.

4.4 Evaluation of the efficiency

In this section, we demonstrate the efficiency of the proposed SA-MDP approach and the COA4ARH algorithm in terms of the running time. In the experiments, we set the number of generations $\mathcal{N}_{gen}$ to 15 and the particle population size $\mathcal{N}_{pop}$ is set to 100. Experimental results are shown in Fig. 5. The detailed analysis is given as follows.

Figure 5 shows the running time of SA-MDP and COA4ARH under three datasets. One can observe that the running time of SA-MDP under three datasets is about half of the running time of COA4ARH. In addition, the SA-MDP approach costs more time to run on Mushroom dataset than on datasets Chess and BMS1.

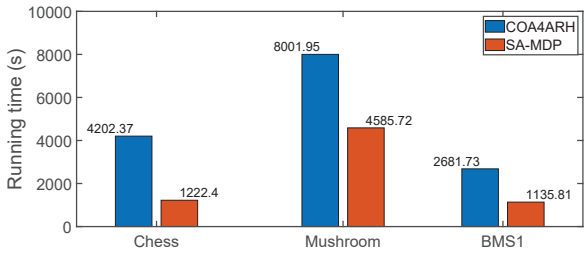


Fig. 5 Running time comparison of SA-MDP and COA4ARH.

Compared with the general EA[18], SA-MDP contains an extra operation: Mother particle splitting. Based on the introduced concept, the old particle can split into several new particles, it improves the diversity of the solution and the exploration ability of SA-MDP. Previous works[19] ignored the utilization of the nature of PSO, i.e., designed to simulate the behavior of flocks of birds. Besides, SA-MDP utilizes that the maximum transactions can be modified and the minimum transactions need to be modified to limit the loss of the data utility. It is different from evolution which just randomly mutates the transactions to get a new mutation result[5]. In such an algorithm, there may be many unnecessary sanitizations. According to Definition 3, fewer modification means less loss NARs. SA-MDP provides an appropriate design for the sanitization, which saves a lot of time consumption.

As a result, we can see that SA-MDP costs a smaller running time than COA4ARH, namely, SA-MDP has a larger convergence rate.

5 Related Work

This section briefly reviews the related works, which can be roughly divided into two folds: perturbation-based data publication and evolutionary algorithm-based mining.

5.1 Perturbation-based data publication

Nowadays, data publication is highly demanded in the industry because it can be utilized to develop data-driven services. The earliest data publication is commonly achieved by data perturbation technologies with privacy-concerned. To perturb the dataset, it consists of two main approaches, i.e., reconstruction and sanitization.

For the reconstruction-based method, the dataset is reconstructed by a certain method and then released to the public[20–22]. By database reconstruction-based technologies, a data publisher can release the dataset for privacy-preserving mining association rules. The reason lies that a published dataset can be partially or fully synthetic based on the requirements of hiding SARs while guaranteeing the utility of the dataset[22]. The kind of methods are interesting and are worth investigating for further researches about the minable data publication. Recently, Li et al.[22] presented a reconstruction-based algorithm DR-PPFIM, which can balance the privacy and the data utility. Specifically, DR-PPFIM algorithm focuses on the sensitive frequent itemsets

and filters out the related frequent itemsets. Then, DR-PPFIM removes these related itemsets. Next, the data publisher performs the reconstruction scheme. Finally, the reconstructed dataset can be released. However, the adopted reconstruction-based method may have a local optimum solution for the minable data publication.

For the sanitization-based method, the dataset removes the sensitive information firstly, and then the dataset is released to the public[23]. There are several works which have been proposed. For instance, some published datasets contain private social media information, which can be discovered by some malicious attackers and be used to predict personal information. Thus, Cai et al.[24] presented a sanitization-based method to decrease the reliability of this kind of attack. Moreover, Liu et al.[25] presented three novel sanitization-based utility mining algorithms to address the privacy problem of sharing data among a variety of organizations.

5.2 Evolutionary algorithm-based mining

Recently, evolutionary algorithm-based mining has also attracted the attention of many researchers because it can improve the efficiency of algorithms. To perturb the dataset, there are two main methods for the evolutionary algorithm-based mining, namely, the single-objective evolutionary algorithm (SOEA)-based method and the multi-objective evolutionary algorithm (MOEA)-based method.

The SOEA-based method has shown its superiority to obtain the optimal solution than traditional methods[26, 27]. For example, 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[12]. It shows better performance than the GA-based hiding strategy[28] and DSR[10]. However, these works focus on how to reduce the privacy leakage risks but ignore the data utility, so the data utility of the protected dataset is low. Therefore, the MOEA-based method has attracted more attention.

There are several MOEA-based methods presented for addressing the problem that can be solved by the SOEA-based method[29–32]. For example, an MOEA-based method in terms of the fuzzy emerging patterns mining was designed[30], a hybrid MOEA method in terms of the rapidly and directly high-quality rules ming was presented[31], and an MOEA with an indexed set representation scheme in terms of the high utility patterns mining was presented[32]. Furthermore, in recent years, many works were presented to solve the privacy-preserving association rules mining problem. For instance, Motlagh and Sajedi[9] designed an MOEA-based algorithm MOSAR, but the computational costs of MOSAR are too high since it hides one SAR at one time. Besides, Cheng[33] investigated the frequent itemsets hiding problem and presented a novel MOEA-based method. The presented method can hide the sensitive frequent itemset by sanitating specific items, but it generates FARs, so the data utility is low.

6 Conclusion

This paper has presented SA-MDP, a novel sanitation approach for achieving the minable data publication while guaranteeing the data privacy of individuals. Essentially, SA-MDP method protects the data privacy via reducing the sensitive information before the data publication. Specifically, to decrease the Support or the Confidence of the SARs, SA-MDP first locates the sensitive items. Then, SA-MDP utilizes the PSO algorithm to find the optimal sanitization method for solving the SARP problem, which is NP-hard. Moreover, PSO enables the particle can learn from the best solution, which guarantees the convergence rate. Besides, SA-MDP adopted the weighted-sum of the side-effects as the fitness function because the data utility and the data privacy are two conflict goals. The presented fitness function can guide the particles to fly towards the optimal solution. In SA-MDP, the particle can split into several child particles, which is achieved by random mutation or learning from the best particle. In this way, the diversity of the population can be increased. In addition, SA-MDP presents a novel preprocessing mechanism before the evolution process of the customized PSO algorithm, filtering out the irreverent transactions, so that the dimension of the search space can be decreased. Finally, we have performed the proposed SA-MDP approach over several datasets. The experimental results have demonstrated the effectiveness and efficiency of SA-MDP. The presented SA-MDP approach hides the SARs by deleting the specific items from the dataset and provides a controllable choice.

Acknowledgment

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

References

[1] D. Su, J. Cao, N. Li, and M. Lyu, PrivPfC: Differentially private data publication for classification, *VLDB J.*, vol. 27, no. 2, pp. 201–223, 2018.

[2] I. 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 Syst. Appl.*, vol. 101, pp. 78–90, 2018.

[3] X. Zheng, G. Luo, and Z. Cai, A fair mechanism for private data publication in online social networks, *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 880–891, 2020.

[4] K. Zhang, Z. Tian, Z. Cai, and D. Seo, Link-privacy preserving graph embedding data publication with adversarial learning, *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 244–256, 2022.

[5] F. Yang, X. Lei, J. Le, N. Mu, and X. Liao, Minable data publication based on sensitive association rule hiding, *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 14, no. 8. pp. 1–11, 2021.

[6] W. Diffie and M. E. Hellman, Special feature exhaustive cryptanalysis of the NBS data encryption standard, *Computer*, vol. 10, no. 6, pp. 74–84, 1977.

[7] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. Mclachlan, A. Ng, B. Liu, P. S. Yu, et al., Top 10 algorithms in data mining, *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[8] M. Li, H. Wang, and J. Li, Mining conditional functional dependency rules on big data, *Big Data Mining and Analytics*, vol. 3, no. 1, pp. 68–84, 2020.

[9] F. N. Motlagh and H. Sajedi, MOSAR: A multi-objective strategy for hiding sensitive association rules using genetic algorithm, *Appl. Artif. Intell.*, vol. 30, no. 9, pp. 823–843, 2016.

[10] S. L. Wang, B. Parikh, and A. Jafari, Hiding informative association rule sets, *Expert Syst. Appl.*, vol. 33, no. 2, pp. 316–323, 2007.

[11] B. Talebi and N. M. Dehkordi, Sensitive association rules hiding using electromagnetic field optimization algorithm, *Expert Syst. Appl.*, vol. 114, pp. 155–172, 2018.

[12] M. H. Afshari, M. N. Dehkordi, and M. Akbari, Association rule hiding using cuckoo optimization algorithm, *Expert Syst. Appl.*, vol. 64, pp. 340–351, 2016.

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

[14] J. Wu, N. Mu, X. Lei, J. Le, and X. Liao, SecEDMO: Enabling efficient data mining with strong privacy protection in cloud computing, *IEEE Trans. Cloud Comput.*, vol. 10, no. 1, pp. 691–705, 2019.

[15] A. Telikani, A. H. Gandomi, and A. Shahbahrami, A survey of evolutionary computation for association rule mining, *Inf. Sci.*, vol. 524, pp. 318–352, 2020.

[16] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, Particle swarm optimization with interswarm interactive learning strategy, *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2238–2251, 2016.

[17] SPMF: An Open-Source Data Mining Library, http://www.philippe-fournier-viger.com/spmf/, 2022.

[18] Z. H. Zhan, S. H. Wu, and J. Zhang, A new evolutionary computation framework for privacy-preserving optimization, in *Proc. 2021 13th Int. Conf. on Advanced Computational Intelligence*, Wanzhou, China, 2021, pp. 220–226.

[19] G. M. Fan and H. J. Huang, A novel binary differential evolution algorithm for a class of fuzzy-stochastic resource allocation problems, in *Proc. 13th IEEE Int. Conf. on Control and Automation*, Ohrid, Macedonia, 2017, pp. 548–553.

[20] I. Dinur and K. Nissim, Revealing information while preserving privacy, in *Proc. of the 22nd ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems*, San Diego, CA, USA, 2003, pp. 202–210.

[21] L. Xu, C. Jiang, J. Wang, J. Yuan, and Y. Ren, Information security in big data: Privacy and data mining, *IEEE Access*, vol. 2, pp. 1149–1176, 2014.

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

[23] A. Telikani and A. Shahbahrami, Data sanitization in association rule mining: An analytical review, *Expert Syst. Appl.*, vol. 96, pp. 406–426, 2018.

[24] Z. Cai, Z. He, X. Guan, and Y. Li, Collective data-sanitization for preventing sensitive information inference attacks in social networks, *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 577–590, 2018.

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

[26] P. 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 Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 3, pp. 324–335, 2020.

[27] I. Fister Jr. and I. Fister, Information cartography in association rule mining, *IEEE Trans. Emerg. Top. Comput. Intell.*, doi: 10.1109/TETCI.2021.3074919.

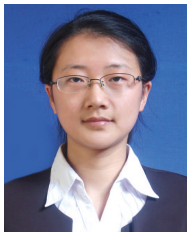
[28] A. Khan, M. S. Qureshi, and A. Hussain, Improved genetic algorithm approach for sensitive association rules hiding, *World Appl. Sci. J.*, vol. 31, no. 12, pp. 2087–2092, 2014.

[29] U. Ahmed, J. C. W. Lin, G. Srivastava, R. Yasin, and Y. Djenouri, An evolutionary model to mine high expected utility patterns from uncertain databases, *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 5, no. 1, pp. 19–28, 2021.

[30] Á. M. García-Vico, C. J. Carmona, P. González, and M. J. del Jesus, MOEA-EFEP: Multi-objective evolutionary algorithm for extracting fuzzy emerging patterns, *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 2861–2872, 2018.

[31] E. V. Altay and B. Alatas, Differential evolution and sine cosine algorithm based novel hybrid multi-objective approaches for numerical association rule mining, *Inf. Sci.*, vol. 554, pp. 198–221, 2021.

[32] 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, *Eng. Appl. Artif. Intell.*, vol. 77, pp. 9–20, 2019.



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

Xiaofeng Liao received the PhD degree in circuits and systems from the University of Electronic Science and Technology of China, Chengdu, in 1997, the BS and MS 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, China. He is currently a professor and the Dean of the College of Computer Science, Chongqing University. From November 1997 to April 1998, he was a research associate with the Chinese University of Hong Kong, Hong Kong, China. From October 1999 to October 2000, he was a research associate with the City University of Hong Kong, Hong Kong, China. 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. He is an associate editor of *IEEE Transactions on Cybernetics* and *IEEE Transactions on Neural Networks and Learning Systems*. He is an IEEE fellow.