A String-Similarity-Oriented Item Swapping Algorithm for Protecting Sensitive Frequent Itemsets in Transaction Databases

Fatah Yasin Al Irsyadi[1], Dedi Gunawan[2], Diah Priyawati[3], Wardah Yuspin[4]
Department of Informatics Engineering-Faculty of Communication and Informatics,
Universitas Muhammadiyah Surakarta, Surakarta, Indonesia[1,2,3]
Department of Law-Faculty of Law and Political Science,
Universitas Muhammadiyah Surakarta, Surakarta, Indonesia[4]

***Abstract*—Frequent itemset mining is a widely adopted data mining technique. The application of the technique can be found in transaction database analysis, such as exploring a set of purchased items. Presently, the growing concern of privacy protection and security issues in society leads the business parties to be more careful in handling their database, since various information can be extracted from the database including the sensitive information. Therefore, database owner should take measure to minimize sensitive information leak during the data mining process. A hiding sensitive frequent itemset algorithm can be adopted to achieve that. However, it is remain a challenge to design a data hiding algorithm that not only successfully hiding frequent sensitive itemset but also minimize the side effects such as item loss, the appearance of artificial frequent itemset and misses cost. In this paper, a method namely D-LSwap that based on item swapping technique is proposed to cope the issue while minimizing those side effects. Initially D-LSwap inspect each transaction in the database to determine whether a transaction is sensitive. Following that, it select a sensitive transaction from the previous process and create a pair of transactions from them. The pair is formed by incorporating Damerau-levensthein string similarity. The next step is selecting items from this pair for the swapping process. Experiment results indicate that the proposed method outperforms several existing algorithms by increasing data utility up to 10%, while minimizing the number of item loss more than 10 times lower than that of the baseline methods.**

***Keywords*—Data hiding; sensitive frequent itemset; frequent itemset; data mining; item swapping technique; D-Lswap**

I. Introduction

Frequent itemset mining in one of the most widely adopted data mining technique that can be found in various domains [1]. Retail business institution frequently employs the frequent itemset mining in transaction database [2]. It perform the task to explore the frequently bought itemset, analyzing customer buying behavior and patterns, as well as analyzing product trend [3]. Even though the analysis is critical, a lot of business institutions facing difficulties due to lack of human capability and computation resources. Furthermore, frequent itemset mining is computationally intensive task [4] that need expert to conduct it. Therefore, they deliver the database to third party to conduct the analysis.

Realizing the fact that the database contains various information including the sensitive information and the growing concern of privacy and laws on data protection drives another challenge in conducting data analysis. Given its significance, privacy is safeguarded through a range of statutory instruments and formal regulations [5], such as the General Data Protection Regulation (GDPR), the Electronic Communications Privacy Act (ECPA), the Children's Online Privacy Protection Rule (COPPA), along with other legal provisions that address particular categories of data. Thus, it becomes a mandatory for the database owner to take measure in minimizing sensitive information from being leaked during the frequent itemset mining process.

One of the strategy to hide the sensitive frequent itemset is called data sanitization. The main concept of data sanitization is hiding sensitive information such as sensitive itemset, while preserving the ability to explore necessary insights from the sanitized databases [6]. Therefore, to achieve that, one needs to transform an original database to a sanitized database to prevent the sensitive frequent itemset from being disclosed. The most widely adopted strategy of hiding sensitive frequent itemset is through item insertion or item suppression procedure in transaction records on the database to reduce the support value of itemset below specific predetermined threshold [7]. The hiding frequent itemset was initially introduced in study [8] and it has been proved to be NP-Hard problem.

The key issue in hiding sensitive frequent itemset hiding is keeping that to be successfully hidden while at the same time minimizing the side effect of the transformation process. The side effects including, excessive data dissimilarity, too much item loss that leads to reducing data utility and high number of newly generated artificial patterns. Various algorithms have been proposed to tackle this issue and those can be categorize into three groups namely, reconstruction-based algorithm, cryptographic-based algorithm and heuristic-based algorithm [9]. Recent finding stated that majority of the proposed methods rely on either item suppression-based or item insertion-based techniques for static transaction database [3]. The main property of these techniques is their ability to minimize the number of newly generated artificial patterns.

Even though the existing methods are effective to hide it, several issues such as excessive item loss, significant data utility loss and high dissimilarity value remain a challenge. Unlike most of the existing works that predominantly build on item suppression or insertion, our proposed method introduces a similarity-guided swapping technique based on the Dam-

TABLE I. ILLUSTRATION OF TRANSACTION DATABASE $\mathcal{D}$

Tid	IID
t_1	1,12,13,18,20
t_2	22,27,28,101
t_3	15,26,37,100,112
t_4	2,13,48,69
t_5	19,32,55,93,100
t_6	42,65,71,95,100,112
t_7	11,13,55,81,100,112
t_8	10,16,45,71,92, 100, 112
t_9	55,112
t_{10}	100,112

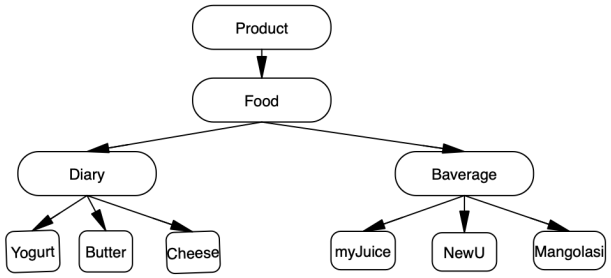

Fig. 1. Item categorization.

erau–Levenshtein similarity. Previous work in [3] also adopts swapping technique where Cosine similarity is applied. However, as described by [10], traditional vector-based similarity measures (e.g., cosine similarity) treat sequences as bags-of-items and thus regardless ordering, which leads to a loss of sequential structure for accurate pattern comparison. Instead, edit-distance measures such as Damerau–Levenshtein better capture sequence transformations relevant for sanitization. Furthermore, the adoption of Damerau-levenstein distance is based on its unique properties where it can calculate transposition and provide more accurate string similarity [11].

The method provides a structure-preserving alternative that maintains sequence length and minimizes semantic distortion. By selecting swap candidates with minimal edit distance, the proposed approach reduces utility loss and improves the fidelity of the sanitized patterns. To the best of our knowledge, no previous work has applied the Damerau-Levenshtein string-similarity metric to design perturbation-based data sanitization. In addition, the method also carefully selects items from the pair to avoid item collision for reducing the number of newly generated artificial patterns. Therefore, this proposal is a novel and meaningful extension to current privacy-preserving data mining literature.

The rest of the paper is organized as follows: Section II describes related work. Section III explains the proposed method. While Section IV and V portray the experiment results and conclusion, respectively.

II. RELATED WORK

The rapid growing of e-commerce and retail industries that collect customer transaction data in daily basis drives business owners to actively extract the hidden information in the databases. To achieve the task, the database owners needs a tool such as data mining algorithms. In addition, the lack of capability and resources to conduct such task for these kind of business institutions directs them to handling the databases to another party such as data miner company. However, due to the growing concern on data security and privacy protection, conducting data mining task becomes more challenging since it can extract sensitive information such as sensitive frequent itemset. Consequently, prior to handle the databases they must take measures to minimize the disclosure of sensitive frequent itemset by hiding it from the databases.

A. Item and Item Category

Transaction databases are constructed from a collection of items I. As can be seen in the transaction database in Fig. I, $I = \{1, 2, 3, 4, 5, ..., 122\}$ where the numbers 1 to 122 represent item's identity number. In practice, items are commonly grouped according to specific criteria or their inherent similarities. Such categorization suggests several advantages, including the buildup of product catalogs and support for marketing programs [12]. Item groupings can be represented using a taxonomy generalization graph, which portrays the hierarchical structure of items, ranging from the most general level to the most detailed. An illustration of this taxonomy generalization graph is presented in Fig. 1.

Each item $i \in I$ belongs to only one category $C = \{C_1, C_2, C_3, ..., C_y\}$ without any overlapping membership to another category. For example, item ID = 10 belongs to C_3 it cannot be owned by other category C_j). The number of category C is determined by the business owners in managing their items. Referring to the Table. I, the collection of items i that exist in a transaction record t is called itemset, where $1 \leq |t_x|$ and $|t| \neq 0$.

B. Frequent Itemset Mining

Frequent itemset mining is one of an essential data mining task that seeks to identify all itemset combinations appearing in transaction records with a frequency greater or equal to a specified threshold [13], [14]. To conduct the task, database owner must define a parameter called minimum support threshold that function to limit occurrence frequency of itemset in the database. Due to there is no universally fixed value exists for this parameter, setting the threshold too low may result in an excessive number of frequent itemsets being generated that might be less informative, whereas setting it too high may cause potentially useful itemsets to be overlooked.

An itemset X is called frequent itemset FI if the $supp(X, \mathcal{D})$ is greater or equal to the number of determined minimum support $minSupp$ [15]. Accordingly, any itemset with a $supp$ value surpassing $minSupp$ can be classified as an FI. The $supp$ value of itemset X in a database $\mathcal{D}$ is computed as in Eq. (1).

$$supp(X, \mathcal{D}) = \frac{f(X)}{|\mathcal{D}|} \tag{1}$$

C. Sensitive Frequent Itemset

A sensitive frequent itemset is a subset of frequent itemsets, denoted as $Fs(X, \mathcal{D}) \subset FI$, whose disclosure may threat the interests of the database owner. Frequent itemsets not classified as sensitive are referred to as non-sensitive itemsets Fn, such

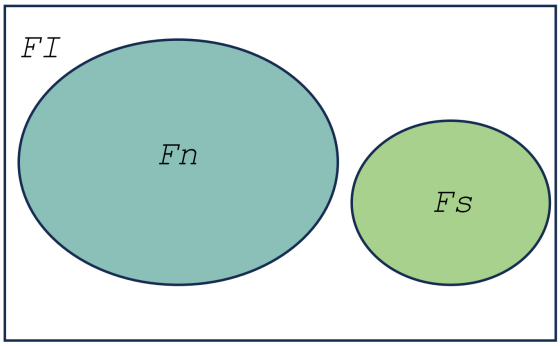

Fig. 2. Relation among Fs, Fn and FI.

that $FI = Fs \cup Fn$ and $Fs \neq Fn$. The relation between Sensitive frequent itemset and frequent itemset can be depicted in Fig. 2.

The pioneering solutions for hiding sensitive frequent itemset were introduced in [8], [16] and [17]. However, these works does not take post modification data quality in to account [7], consequently, the sanitized databases significantly distorted. The proposed method in [18] tries to balance between the privacy protection and data quality in post-modification using one-scan technique. Subsequently, various hiding algorithms were proposed to balance between privacy and data utility and that can be categorized into three groups namely, perturbation-based method, Cryptographic-based method, and Heuristic-based method.

D. Perturbation-Based Method

The perturbation-based method is widely adopted for hiding sensitive frequent itemset due to its flexibility. It relies on injecting noises in the databases either by omitting items or injecting items into a specific transactions. Pioneering work that adopts perturbation-based method was proposed in [19] where one of the proposed solutions is called naive approach. The solution simply removes all item in the sensitive itemsets that exist in the transaction records. While the solution is effective to addresses the hiding sensitive frequent itemset, it results in significant item loss.

Another method that considers various threshold sensitivity has been proposed in study [20]. The method considers the real case in transaction databases where items have certain level of importance for business owners perspective. For example, item a is less valuable than item b since a brings lower profit while item b is more valuable since it generates more profit. To sanitized the databases while reducing the item loss, it creates a template containing a set of victim items that will be omitted from transaction records. A method that adopts local suppression was proposed in [21]. It selectively removing items from sensitive transaction to hide the sensitive frequent itemset with lower items loss. In addition, rotation perturbation method that works for preserving sensitive information in clustering data mining has also been proposed in [22]. To solve the item loss issue, a technique that employs transaction injection has been urges in [23].

Working on frequent itemset mining in transaction database requires high computation resource [24], [25], thus to optimize the performance of data hiding algorithm, a method that utilizes particle swarm optimization (PSO) is proposed in [26]. It generates the sanitized database by excluding sensitive items from certain transaction records while at the same time minimizing the number of item loss. Concerning the issue related to the database size, a method namely MR-OVnTSA is introduced in [27]. The method protect sensitive frequent itemsets by discharge items and transactions for balancing privacy and data utility in big data environment.

1) Cryptographic-based method: High computation resource in hiding sensitive frequent itemset shifts researchers to adopt an efficient solution by using secure multi-party computation. A groundbreaking work in this area is introduced in [28]. The methods use a secure multi-party computation technique where several connected parties perform frequent itemset mining securely. Considering the fact that transaction database has potential to be analyzed by several parties across geographical areas, a scheme of hiding frequent sensitive itemset for distributed system has also been intensively explored [29]. Aiming to enhance the quality of the sanitized database while ensuring the hiding of sensitive frequent itemset, a recent study in [30] adopts cryptographic-based technique to hide sensitive rules in transaction databases. The method strengthen the protection of transaction databases from inference knowledge intrusion. A recent method in [31] employs a cryptographic technique where it improves the mining process by disjoining the encrypted transactions into a certain number of blocks and only uses bilinear pairs of ciphertexts from the blocks. Therefore, the approach becomes more applicable in real-life cases. Ensuring the security of frequent itemset mining, a method successfully introduced differential privacy in two-party scenario [32]. Even though the cryptographic-based method provides a strong privacy guarantee, however, when it meets a huge-sized transactional database, the performance computational efficiency remains a challenge.

2) Heuristic-based method: Since accomplishing both maximum privacy protection and maximum data utility is recognized as an NP-hard problem [33], practical solutions often rely on heuristic-based approaches to address real-world challenges. Numerous heuristic techniques have been introduced under various settings and parameters. Early pioneering efforts in this field include those suggested in [19], [34]. In the existing literature, most heuristic methods employ either item pruning or artificial transaction insertion strategies to lower the support of specific itemsets, thereby effectively concealing sensitive frequent itemsets within a database.

Unlike the earlier approaches, [35] has introduced a distinct scheme that avoids reducing itemset support to conceal sensitive frequent itemsets. Instead, their method focuses on representative rules and eliminates them at the outset. Similarly, [36] proposed a heuristic-based data sanitization approach that relies on item pruning to successfully hide sensitive itemsets within a database. In this method, the selection of items for pruning is guided by calculating the frequency of sensitive items and removing those whose elimination results in minimal item loss.

It is known that heuristics-based strategies tend to diminish the utility of the database, since the removal of items leads to the loss of important information. In addition, the artificial

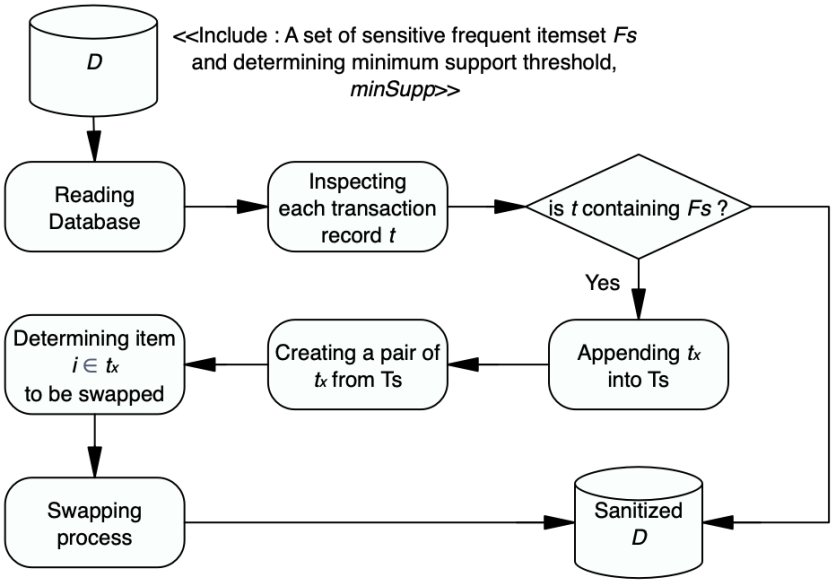

Fig. 3. Proposed swapping method.

transaction insertion approach often induces substantial modifications, causing the composition of items in the sanitized database to differ significantly from that of the original one.

III. Proposed Method

Our proposal is designed for static transaction databases scenario, where the data owner delivers his/her transaction data $\mathcal{D}$ to an external party for conducting frequent itemset mining. The database $\mathcal{D}$ contains a set of transaction record T, where $T = \{t_1, t_2, t_3, ..., t_y\}$. Each transaction record t_x containing a subset of item I, where $I = \{i_1, i_2, i_3, ..., i_m\}$, thus $t_x \subset I$. Referring to the I transaction records $t_3, t_6, t_7, t_8, t_{10}$, all of these records containing $\{100, 112\}$. For example, the itemset $\{100, 112\}$ is referred as to sensitive frequent itemset from the database owner's view. Therefore, when the database $\mathcal{D}$ is released to external party for mining purpose, the $\mathcal{D}$ should be modified in such a way the sensitive frequent itemset mining cannot be exposed in the sanitized database $\widetilde{\mathcal{D}}$.

To generate a sanitized database that can successfully hide the sensitive frequent itemset while at the same time minimizing iside effects such as item loss, data utility loss, artificial patterns and data dissimilarity. A method that follows the swapping technique is suggested. A pioneering research that applied swapping technique is firstly suggested in [37] to prevent disclosure attack on databases. The swapping strategy conveys several unique properties where it neither deletes items from a database nor injects new artificial items or transactions into the database; instead, it swaps items from a transaction to another. Accordingly, the side effects such as item loss, and the data dissimilarity value can be minimized. Our strategy also adopts the Damerau-Levenshtein string similarity algorithm to determine transaction records that are suitable for the swapping process. It does not only provide more accuracy in measuring similarity, but also it can picture the number of edit changes of swapping. The framework of our proposed scheme is depicted in Fig. 3.

An initial step to hide the sensitive frequent itemset is determining the sensitive frequent itemset Fs from the database $\mathcal{D}$. In principal, there are two common approaches to determining the sensitive frequent itemset Fs. The first is allowing database owners to specify sensitive itemsets based on their business objectives, while the second is encouraging customers to classify their purchased items as either sensitive or non-sensitive [38]. In this paper, we follow the first approach, where the database owner designates certain itemsets as the sensitive frequent itemset according to their own perspective and interests. The Fs where $Fs = \{s_1, s_2, s_3, ..., s_k\}$ is composed by a set of sensitive items $i_s \subset I$ and each s_j is a non-empty set of itemset ($|s_j| \geq 1$). Meanwhile, frequent itemset that are not included in Fs is called non-sensitive frequent itemset Fn. The correlation among frequent itemset FI, Fs and Fn is depicted in Fig. 2.

The process of data sanitization also requires prerequisites input by user like minimum support ($minSupp$)threshold and there is no fixed value to determine this. Once the prerequisites value is determine, the nex process is scanning and reading all the transaction t in database $\mathcal{D}$. If the algorithm finds that $t_x \in \mathcal{D}$ containing s_j, $s_j \subseteq t_x$, the t_x is considered as to sensitive transaction and it is appended to Ts. Otherwise, the t_x is appended to Tn. Therefore, only Ts will be processed for the next step in sanitization process. Algorithm 1 represents the pseudo-code of this process.

Once the algorithm collects all the sensitive transaction Ts, the following step is select one of the transaction $t \in Ts$ as a candidate for transaction pairing process, t_a. The selected t_a is then analyzed to obtain its properties such as transaction length which refers to the number of items $i \in t_a$ and item categories C_j that compose it. The algorithm proceed to select another transaction $t \in Ts$, as t_b where this transaction will be the pair of t_a. To obtain the t_b our proposed method applies several criteria. Firstly, to be selected as t_b, the $t \in Ts$ should have the similar item length with that of t_a. This requirement aims to avoid excessive distortion in the transaction record

and maintain the average of transaction length of the database $\mathcal{D}$. Secondly, evaluating item similarity between t_a and each of $t \in Ts$, the most similar one will be selected as t_b. In this step, our method employs Damerau-levenshtein similarity calculation to create the pair. The detail pseudo-code of this step is depicted in Algorithm 2.

Algorithm 1: Reading and Grouping Transaction

Input: $\mathcal{D}$, $i_s \in SI$
Result: Ts and Tn
Scan $\mathcal{D}$
$\forall\ t_x \in \mathcal{D}$
if $s_k \subseteq t_x$ **then**
  add the t_x to Ts
else
  add the t_x to Tn
end

Algorithm 2: Damerau–Levenshtein Distance Algorithm

Input: Two transaction $t_a[1..n]$ and $t_x[1..m] \in Ts$
Output: Edit distance between t_a and t_x, t_b
Initialize matrix $D[0..n, 0..m]$;
for $i \leftarrow 0$ **to** n **do**
  $D[i, 0] \leftarrow i$;
end
for $j \leftarrow 0$ **to** m **do**
  $D[0, j] \leftarrow j$;
end
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow 1$ **to** m **do**
    if $t_a[i] = t_b[j]$ **then**
      $cost \leftarrow 0$;
    else
      $cost \leftarrow 1$;
    end
    $D[i,j] \leftarrow \min\Big\{D[i-1,j]+1,\ \ D[i,j-1]+1,\ \ D[i-1,j-1]+cost\Big\}$;
    if $i > 1$ ***and*** $j > 1$ ***and**** $t_a[i] = t_b[j-1]$ ***and*** $t_a[i-1] = t_b[j]$ **then**
      $D[i,j] \leftarrow \min\{D[i,j], D[i-2,j-2]+1\}$
    end
  end
end
return $t_b \leftarrow Min(D[n,m])$;

Once the pair t_a and t_b is obtained, the proposed method proceed to determine items from both transactions to be swapped each other, this procedure is called victim item Vi selection. The victim item selection incorporates two strategies. The first stage is calculating the frequency f of sensitive items $i_s \in t_a$ that exist in $\mathcal{D}$ and select the one which has the highest occurrence frequency, if there are several sensitive items $i_s \in t_a$ with the same occurrence frequency, the algorithm randomly selects one of them. This strategy allows to minimize the number of item loss since the sensitive items with high frequency will remain observable. The second stage is observing the item category of each $i_s \in t_a$, $i_s \in C_j$. The candidate of Vi should not come from the C_j that has only one item, $|C_j| > 1$. Items $i_s \in t_a$ that satisfies these criteria will be considered as to $Vi(t_a)$.

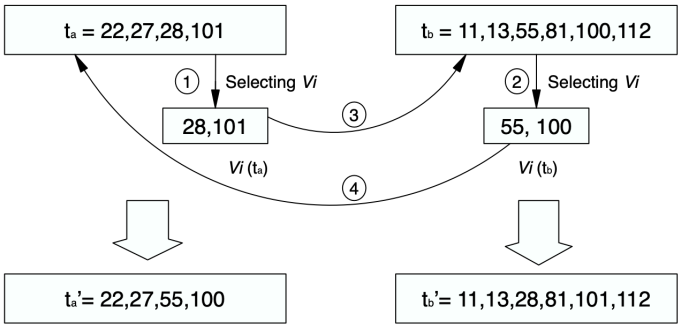

Fig. 4. Procedure of items swapping.

Algorithm 3: Procedure of Vi Selection for Swapping

Input: t_a, t_b
Result: $Vi(t_a)$, $Vi(t_b)$
$\forall\ i_s \in t_a$ calculate the f of $i_s \in t_a$
check the $|C_j| > 1$ is TRUE
$Vi(t_a) \leftarrow Max(f(i_s \in t_a) \& |C_j| > 1$ is **TRUE**
$\forall\ i_s \in t_b$
if $C(Vi(t_a))$ == $C(i_s(t_b))$ **then**
  calculate f of $i_s \in t_a$
  select $Max(f(i_s \int_b))$
  if $i_s \in t_b$ == $i_s \in t_a$ **then**
    $Vi(t_b) \leftarrow i_s \in t_b$
  end
end
return $Vi(t_a), Vi(t_b)$;
end;

The following procedure is determining Vi from t_b. Sensitive items $i_s \in t_b$ can be selected as the $Vi(t_b)$ if it satisfies three requirements. The first requirement is candidate should come from the same category with that of $Vi(t_a)$ to keep the structure of the t remains semantically consistent. The second is candidate should has the highest item frequency among other $i_s \in t_b$ and the third is it does not collide with other items that already exist in t_a. These requirements aims to minimize item loss due to item collision and retain as much as possible the number of items in the $\mathcal{D}$. All these strategies not only guarantee a sanitized database $\widetilde{\mathcal{D}}$ that protects sensitive patterns but also it can minimize the side effects. The detail process of victim item selection is depicted in Algorithm 25.

As the victim items are determined, the last step is swapping them from t_a to t_b and vice versa. Algorithm 31 represents the pseudo-code of the swapping process. In addition to that, as an example we provide the swapping process illustration is expressed in Fig. 4. Supposed the transaction pair $\{t_a, t_b\}$ has been determined where the $t_a = \{22, 27, 28, 101\}$ and $t_b = \{11, 13, 55, 81, 100, 112\}$. The sensitive itemset of t_a is $\{28, 101\}$ while the sensitive itemset of t_b is $\{55, 100\}$. After the evaluation of Vi selection, it is decided that item id $\{28\}$ in t_a is selected and item id $\{55\}$ from t_b is selected

Algorithm 4: Procedure of Items Swapping

Input: $Vi(t_a), Vi(t_b)$, t_a, t_b
Result: $\widetilde{\mathcal{D}}$
create Buffer br_{t_a} and br_{t_b};
br_{t_a}.add $(Vi(t_b)$;
br_{t_b}.add $(Vi(t_a)$;
save to $\widetilde{\mathcal{D}}$;
end;

for the swapping process. The following step is swapping those items from their origin to their destination, thus we can obtain sanitized transactions t'_a and t'_b. Lastly, these transaction records are appended the sanitized database $\widetilde{\mathcal{D}}$.

IV. Experimental Results

$$iLoss = \left| \sum_{i=1}^{d} f\mathcal{D}(i) - \sum_{i=1}^{\widetilde{d}} f\widetilde{\mathcal{D}}(i) \right| \tag{2}$$

$$U(\mathcal{D}, \widetilde{\mathcal{D}}) = \frac{\left| F_{\mathcal{D}} \cap F_{\widetilde{\mathcal{D}}} \right|}{\left| F_{\mathcal{D}} \cup F_{\widetilde{\mathcal{D}}} \right|} \tag{3}$$

$$APR = \frac{|Np \cap Cp|}{|Rp|} \tag{4}$$

$$Diss(\mathcal{D}, \widetilde{\mathcal{D}}) = \frac{1}{\sum_{i=1}^{d} f\mathcal{D}(i)} \times \left| \sum_{i=1}^{d} f\mathcal{D}(i) - \sum_{i=1}^{\widetilde{d}} f\widetilde{\mathcal{D}}(i) \right| \tag{5}$$

Several evaluations to assess the effectiveness of our proposed method is conducted using two real datasets such as the Liquor dataset [39] and BMS-WebView1. Those datasets are most widely adopted datasets in knowledge discovery field. By using two difference datasets we can figure out the performace consistency of the proposed method. Table II summarizes the dataset attributes, and Table III illustrates the corresponding evaluation parameters. We implement the algorithm in Python code and run the system in cloud. A supplementary tool i.e. SPMF from [40] is also adopted to generate frequent itemset by running the FP-Growth algorithm.

In this research four evaluation metrics are adopted to assess our proposed method such as item loss ($iLoss$), data utility ($U(\mathcal{D}, \widetilde{\mathcal{D}})$), artificial pattern ratio (APR), and data dissimilarity ($Diss$). The first metric as denoted in 2 calculates the inequality between the number of items in the $\mathcal{D}$ and that of the sanitized $\widetilde{\mathcal{D}}$. The second metric that stated in 3 computes the amount of data utility that can be preserved in the sanitized database. Specifically, $F_{\mathcal{D}}$ designates the frequent itemsets obtainable from $\mathcal{D}$, and $F_{\widetilde{\mathcal{D}}}$ denotes those frequent itemsets that remain observable in $\widetilde{\mathcal{D}}$.

The third metric as declared in 4 is measuring ratio of the number of artificial pattern or ghost pattern to that of the retained pattern. The measurement is computed by summing the number of newly generated patterns and that of the changed pattern over the number of retained pattern in the sanitized database. The symbols $|Np|$ and $|Cp|$ denote the number of the newly generated frequent itemset and that of the changed frequent itemset pattern in $\widetilde{\mathcal{D}}$, respectively. While $|Rp|$ denotes the number of retained frequent itemset pattern in $\widetilde{\mathcal{D}}$. The last metrics as defined in 5 computes the amount of database dissimilarity after performing data sanitization method. The formula symbols $f\mathcal{D}(i)$ represents the frequency of item i in an original database $\mathcal{D}$ and $f\widetilde{\mathcal{D}}(i)$ refers to that of the sanitized one. In this experiment, we also compare our proposed method (D-LSwap)to several existing method such as $Naive$ [19], $Heuristics$ [36] and random swapping technique (RaS) [41] to investigate its performance.

TABLE II. Datasets Properties

Properties	Datasets	
	BMS-WebView1	**Liquor**
#Transaction	59,602	52,131
#Distinct item	497	4,026
Total items	149,639	410,619
Average length	11.75	8

TABLE III. Testing Parameters

Parameter	Dataset	
	BMS-WebView1	**Liquor**
$minSupp$	0.2% - 0.8%	0.2% - 0.8%
$\|Fs\|$	100	100
Avg. $\|s_k\|$	8	2

A. Item Loss

Depicted in Fig. 5, our proposed method induces lower item loss compared to that of $Naive$ and $Heuristic$ in all $minSupp$ values. These results is mainly due to the $Naive$ and $Heuristic$ approaches perform item suppression strategy that causes more item loss. On the other hand, the RaS has achieved slightly lower item loss to that of our proposed method $D-LSwap$. Such the result is obtained since the RaS algorithm does not consider item category for selecting Vi. Unfortunately, this strategy has drawback in terms of keeping semantical consistency when the items in $\widetilde{\mathcal{D}}$ are generalized in certain analysis. Another experiment using Liquor dataset as shown in Fig. 6 also indicates that the item loss induced by our proposed method achieved almost the same level to that of the SaR algorithm. While the other two methods i.e. $Naive$ and $Heuristic$ promote significant item loss, making it less desirable in scenarios where retention of items is critical. Therefore, it is confirmed that the swapping strategy could minimize the number of item loss compared to the suppression-based strategy.

B. Data Utility

Data utility pictures the number of retained information i.e. frequent itemset obtained from the sanitized database. Referring in Fig. 7, we can inspect the comparison of utility scores across four methods i.e., $D-LSwap$, $Heuristic$, $Naive$, and RaS under varying minimum support thresholds (0.20%, 0.40%, 0.60%, and 0.80%). Overall, $D-LSwap$ consistently achieves the highest utility scores, ranging from 0.9559 at the lowest $minSupp$ value to 0.9334 at the highest, indicating

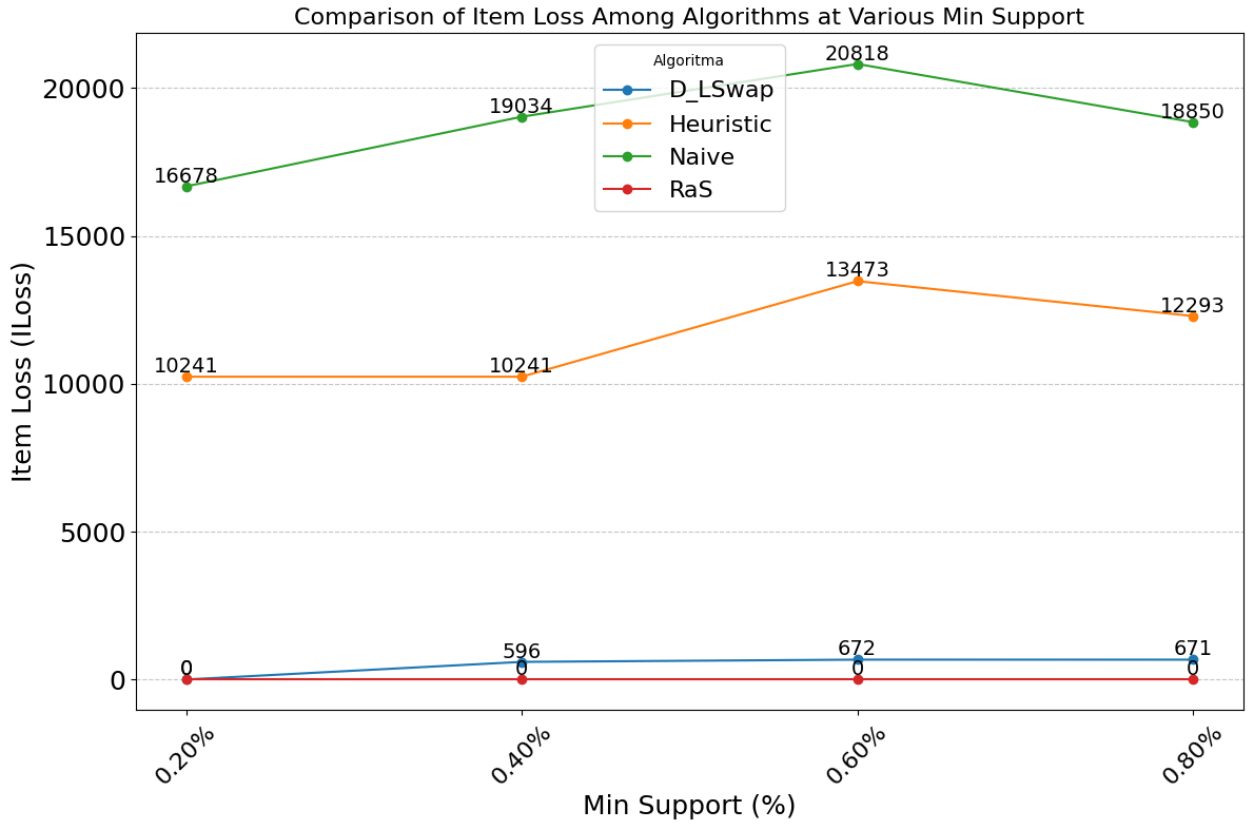

Fig. 5. *iLoss* on BMS-WebView1 dataset.

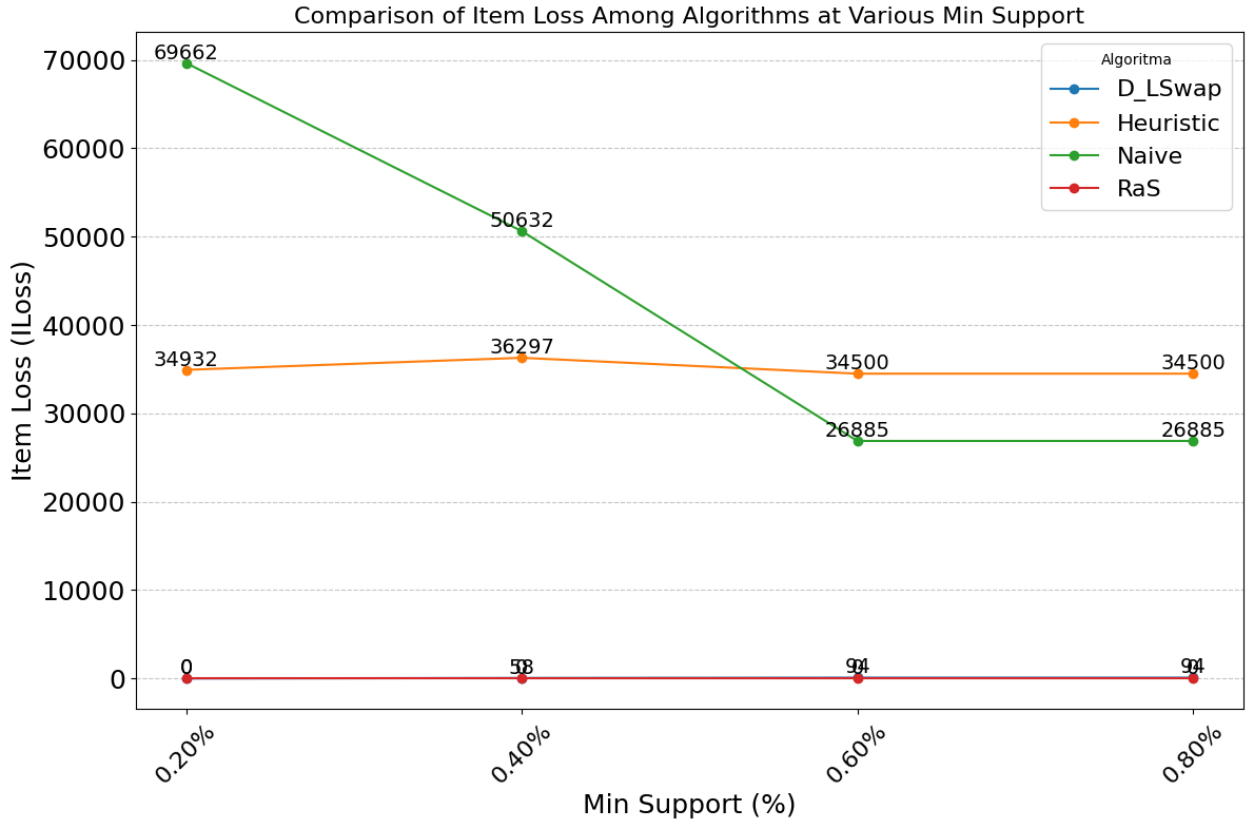

Fig. 6. *iLoss* on Liquor dataset.

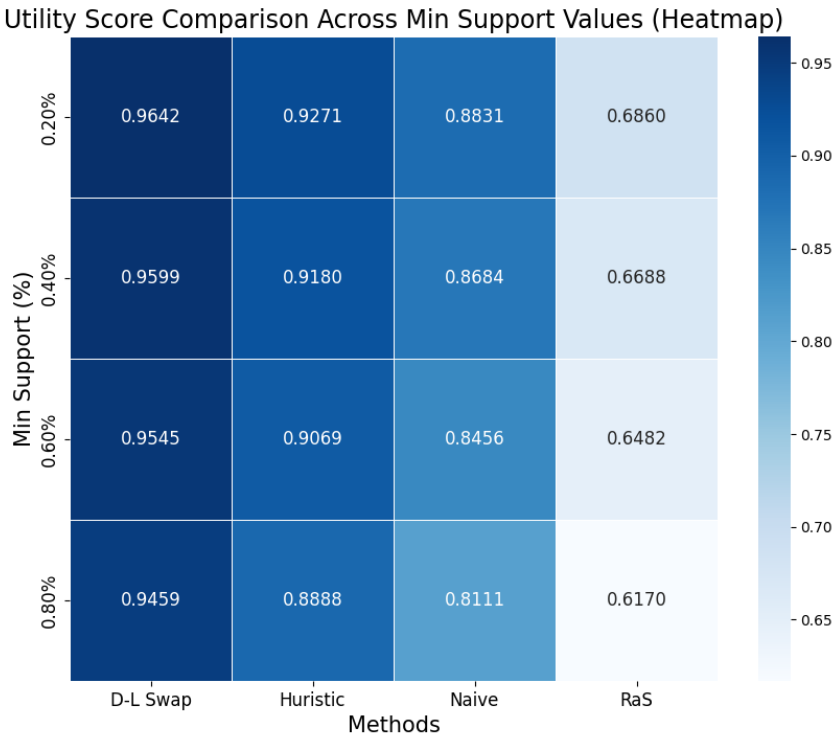

Fig. 7. Data utility on the sanitized BMS-WebView1 dataset.

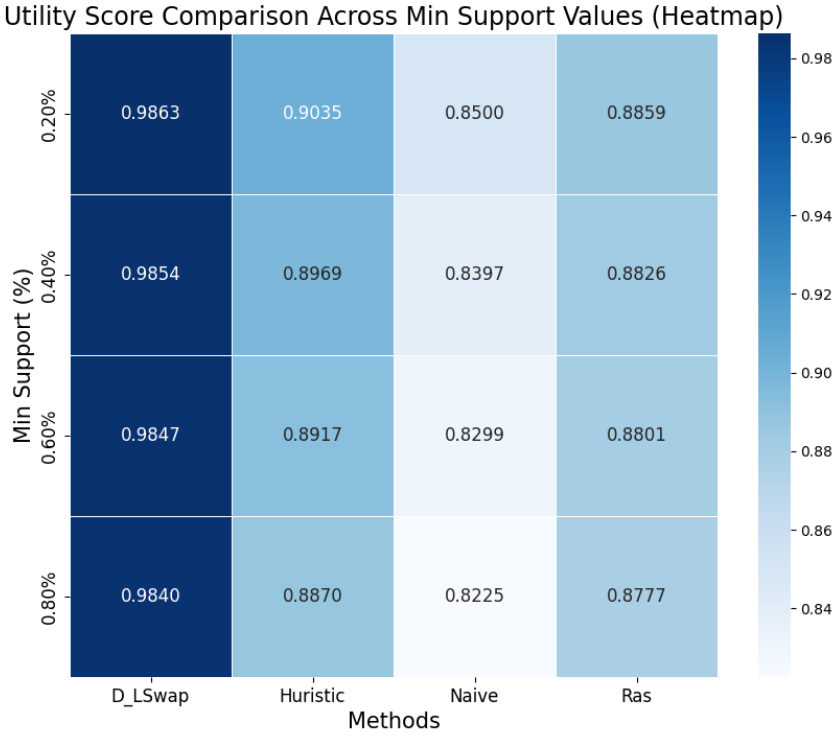

Fig. 8. Data utility on the sanitized Liquor dataset.

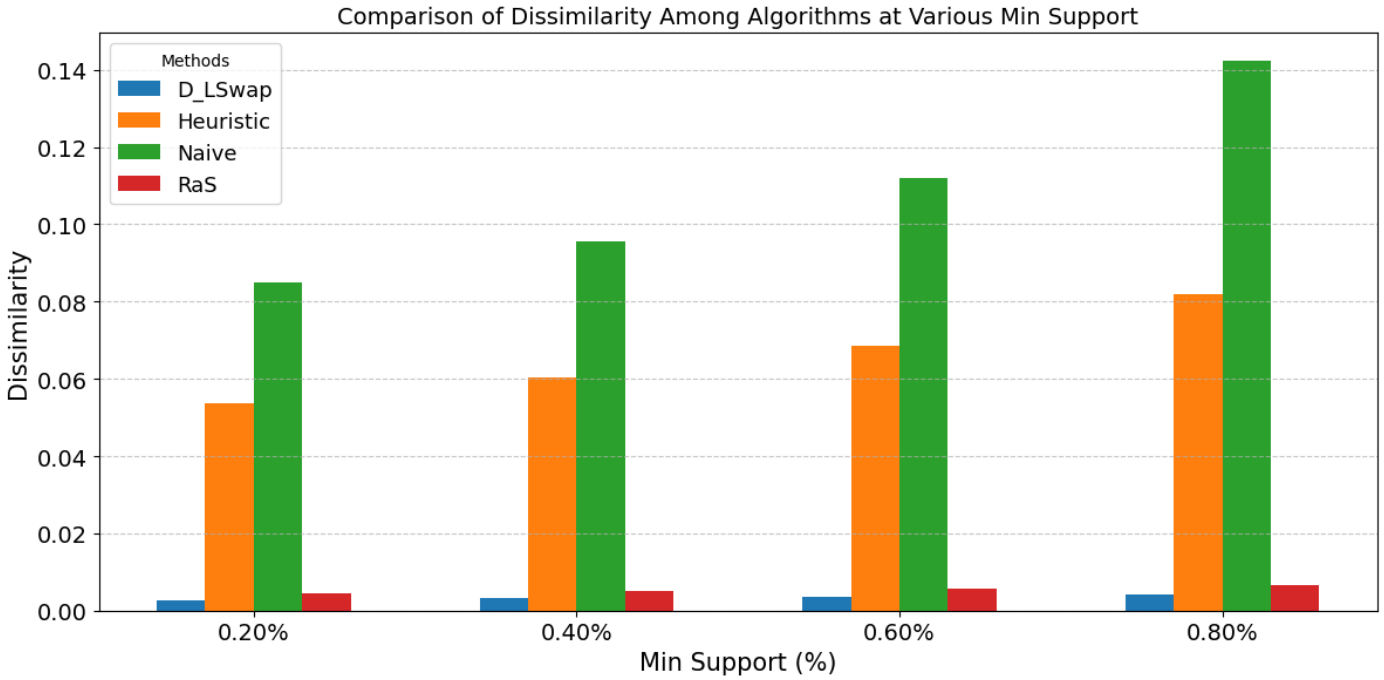

Fig. 9. *Diss* value on BMS-WebView1.

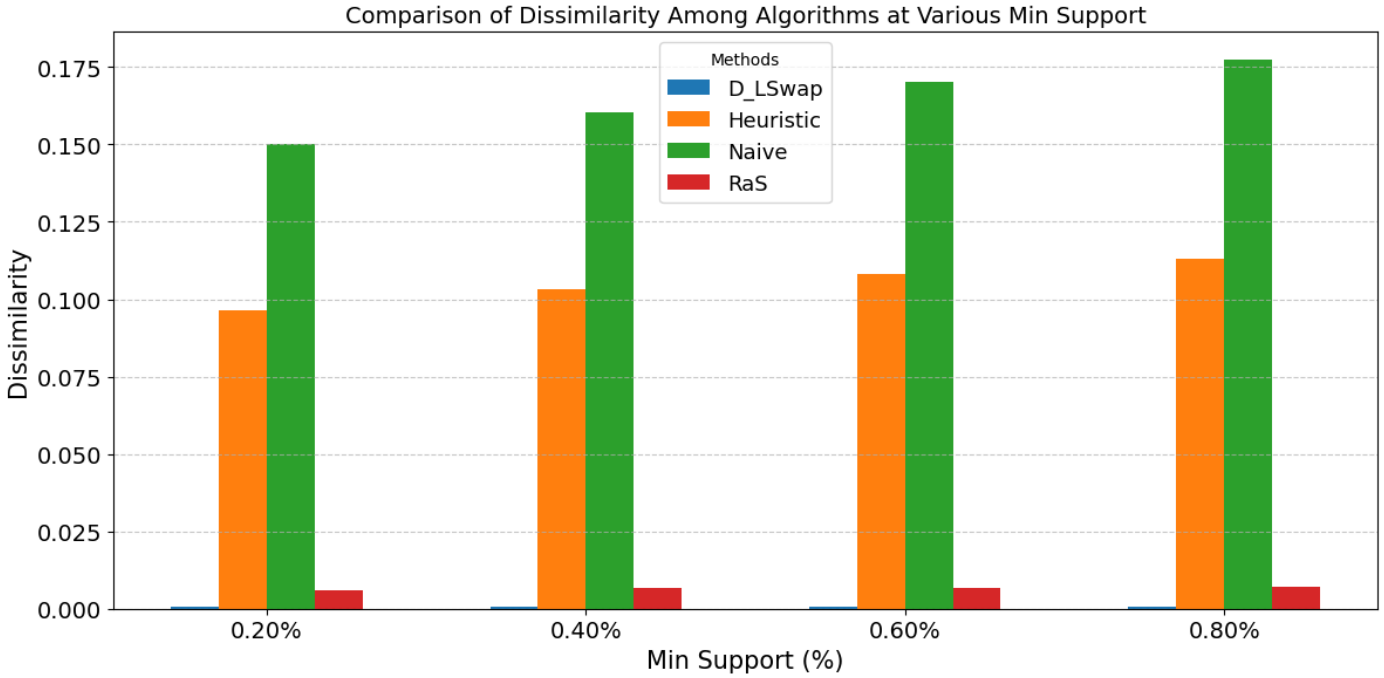

Fig. 10. *Diss* value on liquor.

superior capability in preserving data utility. The $Heuristic$ method follows closely, maintaining relatively high and stable utility scores between 0.9463 and 0.9180. In contrast, the $Naive$ method exhibits a more noticeable decline in utility, dropping from 0.9149 to 0.8576 as the $minSupp$ threshold increases. The RaS demonstrates the lowest performance across all thresholds, with scores falling from 0.7638 to 0.7093, suggesting substantial utility loss regardless of support level. Interestingly, even though both $D-LSwap$ and RaS adopt swapping technique, the resulted data utility is significantly difference since the RaS employs random swapping strategy. The approach causes uncontrolled item disassociation that impact to the itemset correlation in transaction records.

The similar trend also occurs in Fig. 8, our proposed method $D-LSwap$ achieves the highest data utility value in all the $minSupp$ threshold. While in the Liquor dataset $Naive$ approach results in the lowest data utility. The main reason is due to the method applies global suppression procedure to the dataset, consequently each unique item that exist in sensitive frequent itemset are omitted from the dataset.

C. Artificial Pattern Ratio

TABLE IV. ARTIFICIAL PATTERN RATIO IN THE SANITIZED BMS-WEBVIEW

Algorithm	Measurement			
	$\#N_p$	$\#C_R$	$\#Rp$	APR
$D-LSwap$	13	10	87	0.26
$Heuristic$	0	2	82	0.02
$Naive$	5	11	75	0.20
RaS	4	2	89	0.06

TABLE V. ARTIFICIAL PATTERN RATIO IN THE SANITIZED LIQUOR

Algorithm	Measurement			
	$\#N_p$	$\#C_R$	$\#Rp$	APR
$D-LSwap$	68	91	635	0.25
$Heuristic$	2	110	229	0.48
$Naive$	2	172	207	0.84
RaS	105	175	488	0.57

According to the Table IV, Among the evaluated algorithms, $D-LSwap$ exhibits the highest number of artificial patterns (N_P = 13) and the largest APR (0.26), reflecting a relatively greater degree of distortion, although it retains a substantial number of patterns (R_p = 87). In contrast, the Heuristic algorithm introduces no artificial patterns (N_p = 0) and achieves the lowest APR (0.02), suggesting minimal distortion; however, it retains fewer pattern (R_p = 82) compared to $D-LSwap$ and RaS. The $Naive$ algorithm produces the greatest number of changed patterns (C_R = 11) and the lowest pattern retention (R_p = 75), indicating a stronger impact on the original dataset structure. RaS demonstrates a balanced performance, with relatively few artificial patterns (N_p = 4), minimal pattern changes (C_R = 2), the highest pattern retention (R_p = 89), and a low APR (0.06), thereby offering a favorable trade-off between data utility preservation and distortion minimization.

In contrary, the experiment using Liquor dataset as described in Table V suggests that our proposed method can achieve the best results among others. The $D-LSwap$ achieves a relatively moderate number of artificial patterns (N_p = 68) and changed patterns (C_R = 91), while maintaining the highest pattern retention (R_p = 635) and the lowest APR (0.25), suggesting a good balance between pattern preservation and minimal distortion. In contrast, the $Heuristic$ algorithm introduces very few artificial patterns (N_p = 2) however, it shows a higher number of changed patterns (C_R = 110) and a significantly reduced pattern retention (R_p = 229), resulting in a moderate APR (0.48). The $Naive$ algorithm, despite introducing the fewest artificial patterns (N_p = 2), exhibits the highest number of changed patterns (C_R = 172) and the lowest pattern retention (R_p = 207), leading to the highest APR (0.84), which indicates substantial distortion of the original dataset. RaS generates the largest number of artificial patterns (N_p = 105) and changed patterns (C_R = 175), with moderate pattern retention (R_p = 488) and a relatively high APR (0.57). Overall, $D-LSwap$ appears to offer the most favorable trade-off, minimizing distortion while preserving a large proportion of the original rules.

D. Data Dissimilarity

According to the Fig. 9, it can be seen that the proposed method results in the lowest data dissimilarity $Diss$ in all the testing aspects. The results demonstrate that $Naive$ consistently yields the highest dissimilarity values, with a marked increase as the minimum support level rises, peaking at approximately 0.14 at 0.80%. This indicates that $Naive$ introduces the most significant alterations to the original dataset due to its strategy in incorporating suppression approach. $Heuristic$ method exhibits moderate $Diss$ value, which gradually increases with higher support thresholds, suggesting a progressive reduction in data fidelity. In contrast, RaS maintains relatively low $Diss$ across all thresholds, demonstrating a favorable balance between data transformation and preservation. Notably, our proposed method $D-LSwap$ achieves the lowest dissimilarity values overall, remaining nearly constant and close to zero regardless of the support level, which indicates minimal distortion and strong preservation of the original data characteristics.

The similar trend also can be seen in Fig. 10 when the test is conducted using Liquor dataset. $Naive$ consistently exhibits the highest dissimilarity, with values rising from approximately 0.15 at 0.20% support to around 0.18 at 0.80%, indicating substantial alteration to the original data. $Heuristic$ shows moderate dissimilarity, gradually increasing with higher minimum support thresholds, which reflects its progressively more intrusive data modification. RaS demonstrates significantly lower dissimilarity, remaining below 0.01 across all support levels, suggesting minimal impact on the dataset. Finally, D_LSwap consistently achieves the lowest dissimilarity values, staying near zero regardless of the support level, highlighting its strong capability to preserve data similarity. These results confirm that D_LSwap is the most effective approach for minimizing dissimilarity, followed by RaS as a second-best option. Conversely, $Naive$ and $Heuristic$ introduce higher levels of data modification, which may be undesirable in applications prioritizing data fidelity.

According to the experiment in both datasets, we can highlight that the D_LSwap is the most effective algorithm for minimizing dissimilarity, followed by RaS as a secondary option for scenarios where slightly higher modification is

TABLE VI. COMPUTATION COST FOR BMS-WEBVIEW1 DATASET

Method	CPU time sys (s)	Peak memory (MiB)	Memory increment (MiB)
$D - LSwap$	0.3	0.85	0.04
RaS	24.75	10.47	10.46
$Heuristic$	5.48	5.08	0.04
$Naive$	19.18	24.34	12.92

TABLE VII. COMPUTATION COST FOR LIQUOR DATASET

Method	CPU time sys (s)	Peak memory (MiB)	Memory increment (MiB)
$D - LSwap$	0.56	0.77	0.04
RaS	44.56	19.75	19.74
$Heuristic$	35.68	6.81	0.16
$Naive$	66.67	57.77	25.82

acceptable. Conversely, $Naive$ and $Heuristic$ may be less suitable when data fidelity is a primary concern, as they introduce considerably higher levels of distortion.

E. Computational Evaluation

Tables VI and Table VII present the computation cost of four algorithms such as $D - LSwap$, RaS, $Heuristic$, and $Naive$ on two datasets: BMS-WebView1 and Liquor. For the BMS-WebView1 dataset, $D - LSwap$ exhibits the lowest computational cost, requiring only 0.3 seconds of CPU time, 0.85 MiB of peak memory, and a negligible memory increment of 0.04 MiB. In contrast, RaS demands significantly higher resources, with a CPU time of 24.75 seconds and a memory increment of 10.46 MiB. The $Heuristic$ consumes 5.48 seconds of CPU time and a moderate memory footprint (5.08 MiB), while $Naive$ is the most resource-intensive, taking 19.18 seconds and peaking at 24.34 MiB of memory.

A similar trend is observed in the Liquor dataset, where $D - LSwap$ again achieves the best efficiency with only 0.56 seconds of CPU time, 0.77 MiB of peak memory, and 0.04 MiB memory increment. The $Naive$ method records the highest computational overhead with 66.67 seconds of CPU time and 57.77 MiB of peak memory. RaS and $Heuristic$ require 44.56 seconds and 35.68 seconds, respectively, with moderate memory usage compared to $Naive$. These results demonstrate that $D-LSwap$ is consistently the most efficient algorithm, achieving minimal runtime and memory overhead across both datasets. This efficiency makes $D-LSwap$ highly suitable for large-scale or resource-constrained environments, while $Naive$ appears impractical due to its high computational cost.

V. DISCUSSION

This research proposed a data hiding method that based on swapping technique. Experiment results highlight that swapping technique can successfully hide sensitive information and it can significantly reduce the number of item loss and minimize the data dissimilarity value compared to that of suppression based techniques. This finding support the previous research from [42] since the swapping technique is neither remove nor add items from databases. The swapping-based technique allows items in the dataset remain observable and it is beneficial for data analytic that regards the presence of items in the database is crucial.

In contrary, since pairing transaction is performed by scanning each transaction while computing its similarity values, it may takes significant computing resource as the number of the sensitive transaction grows. Additionally, due to some items from a transaction are swapped to another transaction, the structure or composition of the swapped transactions can change drastically, and therefore, it may generate several new rules that previously do not exist in databases. Applying the swapping-based technique should also be carefully considered in health-related dataset since it may generate false information that threat people's life.

VI. CONCLUSION

In this work, a framework is introduced for protecting sensitive frequent itemsets in transactional databases, thereby reducing the likelihood of sensitive data leakage. The proposed method performs item swapping strategy by adopting Damerau-Levensthein string similarity algorithm. It creates a pair of transaction using the similarity algorithm and determines items from those two transaction to perform swapping. Therefore, it allows the privacy is kept protected, with minimizing side effects such as minimizing the number of item loss, maintaining data utility, reducing data dissimilarity and keeping the properties of the sanitized dataset remains similar to that of the original one.

The experiment results demonstrate that $D - LSwap$ consistently outperforms the other methods in terms of maintaining the same pattern, dissimilarity minimization, and computational efficiency. $Naive$, while simple, incurs high computational cost and significantly distorts the data, making it less suitable for scenarios requiring high data utility. $Heuristic$ and RaS achieve moderate performance in terms of both utility preservation and efficiency, even so our proposed method outperformed it. These findings suggest that $D - LSwap$ can be a reliable solution for privacy-preserving data publishing for business institutions prior to share their customer transaction data to external parties.

Further exploration in the use of other string similarity algorithms is encouraged to obtain the best pair of transaction records for swapping. In addition, a profound strategy such as using optimization technique for selecting items also necessary for future research to achieve a balance between maintaining both accuracy and efficiency.

ACKNOWLEDGMENT

This research is fully funded by the Directorate of Research and Community Service Ministry of Higher Education, Science, and Technology of the Republic of Indonesia, under the Fundamental Research scheme with grant number 127/C3/DT.05.00/PL/2025, 007/LL6/PL/AL.04/2025, 168.23/A.3-III/LRI/VI/2025.

REFERENCES

[1] L. Bustio-Martínez, R. Cumplido, M. Letras, R. Hernández-León, C. Feregrino-Uribe, and J. Hernández-Palancar, "Fpga/gpu-based acceleration for frequent itemsets mining: A comprehensive review," *ACM Comput. Surv.*, vol. 54, no. 9, Oct. 2021. [Online]. Available: https://doi.org/10.1145/3472289

[2] T. Y. Prawira, S. Sunardi, and A. Fadlil, "Market Basket Analysis To Identify Stock Handling Patterns & Item Arrangement Patterns Using Apriori Algorithms," *Khazanah Informatika : Jurnal Ilmu Komputer dan Informatika*, vol. 6, no. 1, pp. 33–41, 2020.

[3] D. Gunawan, Y. S. Nugroho, and Maryam, "Swapping-based Data Sanitization Method for Hiding Sensitive Frequent Itemset in Transaction Database," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, pp. 693–701, 2021.

[4] M. Sadeequllah, A. Rauf, S. U. Rehman, and N. Alnazzawi, "Probabilistic Support Prediction: Fast Frequent Itemset Mining in Dense Data," *IEEE Access*, vol. 12, no. February, pp. 39 330–39 350, 2024.

[5] W. Yuspin, H. A. Afnan, K. Wardiono, A. Budiono, A. L. Prakoso, T. Rajput, A. B. Bal, and J. Pitaksantayothin, "Deep Fakes in P2PL Services: Assessing Legal Challenges and Data Privacy Risks," *WSEAS Transactions on Computer Research*, vol. 13, pp. 469–479, 2025.

[6] U. Ahmed, G. Srivastava, and J. C. W. Lin, "A Machine Learning Model for Data Sanitization," *Computer Networks*, vol. 189, no. November 2020, p. 107914, 2021. [Online]. Available: https://doi.org/10.1016/j.comnet.2021.107914

[7] V. S. Verykios, E. C. Stavropoulos, P. Krasadakis, and E. Sakkopoulos, "Frequent itemset hiding revisited: pushing hiding constraints into mining," *Applied Intelligence*, 2021.

[8] M. Atallah, A. Elmagarmid, M. Ibrahim, E. Bertino, and V. Verykios, "Disclosure Limitation of Sensitive Rules," in *Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange*, ser. KDEX '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 45—-. [Online]. Available: http://dl.acm.org/citation.cfm?id=519168.788219

[9] D. Gunawan, "Classification of Privacy Preserving Data Mining Algorithms : A review," *Jurnal Elektronika dan Telekomunikasi (JET)*, vol. 20, no. 2, pp. 36–46, 2020.

[10] K. Rieck, "Similarity measures for sequential data," *Wiley Int. Rev. Data Min. and Knowl. Disc.*, vol. 1, no. 4, pp. 296–304, jul 2011. [Online]. Available: https://doi.org/10.1002/widm.36

[11] K. Kleshch and V. Shablii, "Comparison of Fuzzy Search Algorithms Based on Damerau-Levenshtein Automata on Large Data," *Technology Audit and Production Reserves*, vol. 4, no. 2, pp. 27–32, 2023.

[12] D. Shen, J. D. Ruvini, M. Somaiya, and N. Sundaresan, "Item Categorization in the e-Commerce Domain," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, ser. CIKM '11. New York, NY, USA: ACM, 2011, pp. 1921–1924. [Online]. Available: http://doi.acm.org/10.1145/2063576.2063855

[13] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 2012.

[14] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 2016.

[15] X. Cheng, S. Su, S. Xu, P. Tang, and Z. Li, "Differentially private maximal frequent sequence mining," *Computers and Security*, 2015.

[16] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino, "Hiding association rules by using confidence and support," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2001.

[17] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni, "Association Rule Hiding," *IEEE Transactions on Knowledge and Data Engineering*, 2004.

[18] S. R. M. Oliveira and S. R. M. Oliveira, "An E cient One-Scan Sanitization For Improving The Balance Between Privacy And Knowledge Discovery," *Computing*, no. June, 2003.

[19] S. Oliveira and O. Zaiane, "Privacy preserving frequent itemset mining," *Proceedings of the IEEE international conference on Privacy, security and data mining-Volume 14*, vol. 14, pp. 43–54, 2002. [Online]. Available: http://portal.acm.org/citation.cfm?id=850782.850789

[20] Y. P. Kuo, P. Y. Lin, and B. R. Dai, "Hiding frequent patterns under multiple sensitive thresholds," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5181 LNCS, pp. 5–18, 2008.

[21] R. Chen, B. C. M. Fung, N. Mohammed, B. C. Desai, and K. Wang, "Privacy-preserving trajectory data publishing by local suppression," *Information Sciences*, vol. 231, pp. 83–97, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2011.07.035

[22] K. Chen and L. Liu, "Privacy preserving data classification with rotation perturbation," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, 2005, pp. 4 pp.–.

[23] L. Chun-Wei, H. Tzung-Pei, C. Chia-Ching, and W. Shyue-Liang, "A Greedy-based Approach for Hiding Sensitive Itemsets by Transaction Insertion," *Journal of Information Hiding and Multimedia Signal Processing.*, vol. 4, no. 4, pp. 201–2014, 2013.

[24] S. Kim, "Optimizing Privacy in Set-Valued Data: Comparing Certainty Penalty and Information Gain," *Electronics (Switzerland)*, vol. 13, no. 23, 2024.

[25] Y. Xun, J. Zhang, H. Yang, and X. Qin, "HBPFP-DC: A parallel frequent itemset mining using Spark," *Parallel Computing*, vol. 101, p. 102738, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167819120301198

[26] S. Jangra and D. Toshniwal, "VIDPSO: Victim item deletion based PSO inspired sensitive pattern hiding algorithm for dense datasets," *Information Processing and Management*, vol. 57, no. 5, p. 102255, 2020. [Online]. Available: https://doi.org/10.1016/j.ipm.2020.102255

[27] S. Sharma and D. Toshniwal, "MR-OVnTSA: a heuristics based sensitive pattern hiding approach for big data," *Applied Intelligence*, vol. 50, no. 12, pp. 4241–4260, 2020.

[28] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2000.

[29] S.-y. Kuno, K. Doi, and A. Yamamoto, "Frequent closed itemset mining with privacy preserving for distributed databases," in *2010 IEEE International Conference on Data Mining Workshops*, 2010, pp. 483–490.

[30] N. Rajesh and A. A. L. Selvakumar, "Association rules and deep learning for cryptographic algorithm in privacy preserving data mining," *Cluster Computing*, vol. 22, no. s1, pp. 119–131, 2019. [Online]. Available: https://doi.org/10.1007/s10586-018-1827-6

[31] C. Ma, B. Wang, K. Jooste, Z. Zhang, and Y. Ping, "Practical Privacy-Preserving Frequent Itemset Mining on Supermarket Transactions," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1992–2002, 2020.

[32] W. Chen, H. Chen, T. Han, W. Tong, and S. Zhong, "Secure two-party frequent itemset mining with guaranteeing differential privacy," *IEEE Transactions on Mobile Computing*, vol. 24, no. 1, pp. 276–292, 2025.

[33] H. Liang and H. Yuan, "On the Complexity of t-Closeness Anonymization and Related Problems BT - Database Systems for Advanced Applications," W. Meng, L. Feng, S. Bressan, W. Winiwarter, and W. Song, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 331–345.

[34] A. HajYasien and V. Estivill-Castro, "Two new techniques for hiding sensitive itemsets and their empirical evaluation," *Data Warehousing Knowledge Discovery, Proc.*, vol. 4081, pp. 302–311, 2006.

[35] D. Jain, P. Khatri, R. Soni, and B. K. Chaurasia, "Hiding sensitive association rules without altering the support of sensitive item(s)," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 84, no. PART 1, pp. 500–509, 2012.

[36] D. Gunawan and L. Guanling, "Heuristic Approach on Protecting Sensitive Frequent Itemsets in Parallel Computing Environment," in *The 1ST UMM International Conference on Pure and Applied Research (UMM-ICOPAR 2015)*, Malang, East Java, Indonesia, 2015, pp. 41–49.

[37] T. Dalenius and S. P. Reiss, "Data-swapping: A technique for disclosure control," *Journal of Statistical Planning and Inference*, vol. 6, no. 1, pp. 73–85, 1982. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0378375882900581

[38] T. P. Hong, C. W. Lin, K. T. Yang, and S. L. Wang, "Using TF-IDF to hide sensitive itemsets," *Applied Intelligence*, vol. 38, no. 4, pp. 502–510, 2013.

[39] P. Fournier-Viger, "Foodmart dataset," 2020. [Online]. Available: http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php

[40] P. Fournier-Viger, J. C. W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam, "The SPMF open-source data mining library version 2," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9853 LNCS, pp. 36–40, 2016.

[41] S. E. Fienberg and J. McIntyre, "Data swapping: Variations on a theme by dalenius and reiss," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3050, pp. 14–29, 2004.

[42] D. Gunawan and M. Mambo, "Data anonymization for hiding personal tendency in set-valued database publication," *Future Internet*, vol. 11, no. 6, 2019.