PTA: An Efficient System for Anonymizing Transaction Databases

Jerry Chun-Wei Lin, Qiankun Liu, Philippe Fournier-Viger, and Tzung-Pei Hong

Abstract—Several approaches have been proposed to anonymize relational databases with k-anonymity, to avoid the disclosure of sensitive information by re-identification attacks. A relational database is said to meet the criteria of k-anonymity if each of its records is identical to at least (k-1) other records with respect to quasi-identifier attributes. To attain the goal of k-anonymity for a transactional database, each item must successively be considered as a quasi-identifier attribute, but this greatly increases dimensionality, the computational complexity of anonymization, and the loss of information resulting from the anonymization process. In this paper, an efficient anonymization system called PTA is proposed to not only anonymize transactional data with a small information loss but also to reduce the computational complexity of the anonymization process. It consists of three modules such as a pre-processing module, a Travelling Salesman Problem (TSP) module, and an anonymization module to anonymize the transaction data and guarantees that at least k-anonymity is achieved. Extensive experiments have been carried to compare the efficiency of the designed approach with the state-of-the-art algorithms for anonymity in terms of scalability of the proposed divide-and-conquer mechanism, runtime, and information loss. Results indicate that the proposed PTA system outperforms the compared algorithms in all respects.

Index Terms—anonymity; TSP; divide-and-conquer; Gray sort; privacy preserving.

1 Introduction

In recent years, transaction databases have attracted a lot of interest from researchers due to their many real-life applications. Several techniques related to data mining [1, 2], recommendation systems [3, 4], and web search personalization [5, 6] have been developed to utilize or analyze information stored in transaction databases. Since transactional data is collected about all aspects of people's lives, it may contain sensitive personal information such as information about sexual orientation, religion, medical conditions, and social insurance numbers. As a result, if private information stored in transaction databases is accessed by attackers, sensitive or confidential information may be revealed, which may lead to serious security threats such as identity theft. Moreover, even if a database is anonymized (e.g. by removing names), sensitive information may be inferred from non-sensitive information if that information is different for many persons.

In recent decades, k-anonymity [7] and l-diversity [8] have become important research topics as they provide criteria that should be met to publish data, while ensuring the preservation of privacy. The main goal of k-anonymity is to prevent re-identification attacks on a relational database, that is to ensure that individuals in an anonymized database cannot be re-identified based on their non-sensitive attribute values. A relational database is said to meet the criteria of k-anonymity if each of its records is identical to at least (k-1) other records with respect to quasi-identifier attributes. Most of the anonymization techniques based on k-anonymity generate equivalence classes of k records using Domain Generalization Hierarchies of quasi-identifier attributes such as sex, birth date, and zip code [9, 10, 11]. Records in each equivalence class are then transformed so that they all have identical quasi-identifier attribute values. Since an equivalence class contains k non-distinguishable records, the probability that an attacker successfully identify a person is no more than $1/k$ when using k-anonymity [12, 13, 14].

Besides the anonymization of relational data, algorithms [15, 16, 17, 18] proposed in previous studies are mainly designed to reduce information loss caused by anonymization. Hence, these approaches remains in some cases vulnerable to re-identification attacks. In this paper, we aim at not only preventing the disclosure of sensitive information in transactional data but also to reduce information loss and the time required to perform anonymization for high dimensional transactional data. An anonymization system

- *Jerry Chun-Wei Lin and Qiankun Liu are with the School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China.*
  E-mail: jerrylin@ieee.org; mcqueenqkliu@gmail.com
  Website: http://ikelab.net
- *Philippe Fournier-Viger is with the School of Natural Sciences and Humanities, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China.*
  E-mail: philfv@hitsz.edu.cn
- *Tzung-Pei Hong is with the Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan.*
  E-mail: tphong@nuk.edu.tw

named PTA is proposed, which consists of three modules named the **P**re-processing module, the **T**SP module, and the **A**nonymity module. The proposed system guarantees *k*-anonymity for transactional data. The major contributions of this paper are summarized as follows.

1) The challenges raised by the high-dimensionality of transactional data for anonymization are addressed in the developed system by using a divide-and-conquer approach to partition transactions into several segments based on the Hamming distance. Each segment is then individually processed by a designed algorithm named PrimTSP to find the shortest cyclical path that reduces the loss of information for anonymization. Because the data is partitioned into segments that are processed independently, the cost of finding the shortest path is greatly reduced.
2) The PrimTSP algorithm is proposed to find a shortest cyclical path as a local approximate solution to the Traversal Salesman Problem (TSP). This process is performed to find the most similar consecutive transactions in the database. Furthermore, transactions are sorted by the Gray order to achieve global optimization of information loss in segments.
3) A new mapping and majority-voting process is developed to find the most similar transactions in a segment. Those are then assigned to the same group based on a symmetric mapping approach. A majority-voting approach is then employed to find the center of the group. Then, groups having the least information loss are selected as equivalence classes and all transactions in each equivalence class are replaced by the same center point. Based on the above ideas, the propose system minimizes information loss.

2 Related Work

Preserving privacy when publishing data is crucial because otherwise published information may be used to identify persons, and as a result critical personal information may be revealed. Numerous algorithms have been proposed to anonymize databases [19, 20, 21, 22, 23] to prevent re-identification attacks. A popular criteria for anonymization is *k*-anonymity [9, 7, 11], which states that each record or transaction in a database must be identical to at least (*k*-1) other records, where k is a parameter set by the user. In general, techniques for attaining k-anonymity construct equivalence classes according to Domain Generalization Hierarchies of quasi-identifier attributes such as sex, birth date, or zip code. Many techniques to obtain k-anonymity have been proposed to anonymize databases, including some based on the concepts of generalization [9, 7, 14], suppression [24, 25], clustering [26, 10] and perturbation [11, 16].

Xu et al. [14] proposed a generalization-based algorithm to anonymize relational databases using a recoding approach. This approach calculates the utility of each attribute and considers differences between items. Kisilevich et al. [24] proposed a new *k*-anonymity method that relies on classification trees and suppression to attain anonymity. This approach suppresses an attribute value in a record if it is highly correlated with the values of sensitive attributes, to ensure anonymity. An advantage of this approach is that the user does not need to provide domain hierarchy trees to be used by the generalization process. Abul et al. [26] proposed a novel *k*-anonymity approach based on the concept of co-localization to anonymize databases about moving objects. This approach perturbs the trajectories of moving objects both in terms of space and time to meet the anonymity goal set by the user.

So far, *k*-anonymity techniques have been designed for various types of data. Poulis et al. [27] considered the anonymization of Relational-Transactional-datasets (RT-datasets), where each record contains both relational attributes and transaction items. Two frameworks were proposed to preserve privacy while minimizing information loss in RT-datasets. Doka et al. [28] formulated the problem of maximal-utility *k*-anonymity as a network flow problem to achieve the full potential of heterogeneity and gain higher utility while providing the same privacy guarantee for syntactic data. Wang et al. [29] proposed a novel utility measurement based on graph models. A general *k*-anonymity framework was also designed, which can be used with various utility measures to achieve *k*-anonymity with a small utility loss from social networks. Furthermore, Chettri and Borah [30] proposed a method called Microaggregation based Classification Tree (MiCT) to achieve *k*-anonymity using the methods of generalization and suppression, for privacy preserving classification of data.

Most approaches for attaining *k*-anonymity are designed for relational data, and hence cannot be directly applied to transactional data. The reason is that if all items (attributes) are considered as quasi-identifiers (QIDs), the problem of anonymization has a very high dimensionality and solving such problem is very expensive. Moreover, if a transactional database is too sparse, the information loss as a result of anonymization will be very high since each transaction is highly different from each other in a sparse database [17]. Xu et al. [31] introduced a novel concept to ensure the privacy of transactional data named (*h*, *k*, *p*)-coherence, and designed a greedy algorithm to achieve anonymity while preserving as much sensitive information as possible. Ghinita et al. [32] considered the correlation between purchased products to preserve privacy, such that non-sensitive items cannot be used to infer sensitive information. This approach solves the problem of the high dimensionality of transactional data for anonymization. Wang et al. [17] presented a sensitive *k*-anonymity approach to anonymize sensitive attributes, by ensuring that each transaction has at least (*k*-1) identical transactions. Although, this approach provides anonymity for transactional data, it still produces a high information loss since highly similar transactions are replaced by the center point in each equivalence class. Xue et al. [18] proposed an algorithm to transform each set-valued record into a bitmap for generalizing the QIDs in a non-reciprocal recoding way. This method successfully reduces information loss but increases the risk of privacy disclosure since the original data may not be well-anonymized and hence sensitive information may still be identified. Besides, this approach uses a genetic algorithm to find the set of transactions to anonymize, which can lead to long execution times and has a high complexity. Hsu et al. [33] proposed the *k*-anonymity of multipattern (KAMP) problem to protect data from re-identification using an hybrid approach, which aims at satisfying the *k*-anonymity of individual patterns. Although this approach uses a perturbation technique to hide sensitive information, there remains a risk that individuals may be re-identified.

3 Preliminaries and Problem Statement

A set-valued dataset (a transaction database) is a set of transactions denoted as $D = \{T_1, T_2, \ldots, T_n\}$, where n is the number of transactions. A set $I = \{I_1, I_2, \ldots, I_d\}$ represents all items occurring in the dataset, where $d = |I|$ is the number of distinct items. Furthermore, let there be a set of sensitive items SI denoted as $SI = \{s_1, s_2, \ldots, s_m\}$ such that $SI \subseteq I$. Let the non-sensitive items in I be called the quasi-identifier items, defined as the set $QID = I - SI$, containing $m - d$ items.

TABLE 1: A transaction database.

TID	Symptoms (QIDs)	Sensitive Items (SI)
T_1	a, b	Cardiopulmonary, Cancer
T_2	b, c	AIDS
T_3	b, c, d	Hepatitis, Influenza
T_4	a, b, c	Cancer
T_5	a, b, d	Influenza
T_6	a, c, d	Leukemia, AIDS

The set-valued dataset that will be used as running example to illustrate definitions is shown in Table 1, where each transaction represents a person. This dataset contains non-sensitive items (*QID*) and sensitive items (*SI*). If this dataset is made public or if it is accessed, an important privacy risk is that sensitive information about users may be discovered by malicious persons even if they only partial access to the information contained in the dataset. For example, if a person named Alex knows that his colleague Bob has the symptoms *a*, *b*, and *c*, he can deduce that Bob is the person represented by transaction T_4, and thus know the sensitive information that Bob has cancer.

To prevent the re-identification of transactions in D as in the above example, it is necessary to transform the dataset D into a new anonymized dataset D' that meets the constraint of *k*-anonymity. This later constraint states that each transaction needs to have at least $(k - 1)$ identical transactions. The operations of addition (item generalization) and deletion (item suppression) can be used to achieve this purpose. For example, in Table 1, transactions T_2, T_3, and T_4 could be assigned to the same equivalence class and then each of those transactions could be replaced by a transaction (b, c). Similarly, T_1, T_5, and T_6 could be assigned to a same equivalence class and then those transactions could be replaced by the transaction (a, b, d). The dataset resulting from this transformation is shown in Table 2.

TABLE 2: A dataset respecting the criteria of 3-anonymity.

TID	Symptoms (QIDs)	Sensitive Items (SI)
T_1	a, b, d	Cardiopulmonary, Cancer
T_2	b, c	AIDS
T_3	b, c	Hepatitis, Influenza
T_4	b, c	Cancer
T_5	a, b, d	Influenza
T_6	a, b, d	Leukemia, AIDS

The dataset of Table 2 meets the criteria of 3-anonymity since each transaction is identical to at least $(k - 1)$ (= 2) transactions.

Definition 1 (*k*-anonymity of transactional data)**.** A transaction database D' meets the constraint of *k*-anonymity if every transaction $T_j \in D'$ has at least $(k - 1)$ identical transactions in D'. Thus, the probability that each transaction is re-identified is no more than $1/k$.

To meet the criterion of *k*-anonymity, an anonymization algorithm modifies transactions containing sensitive information in a database so that each of them is at least identical to $k - 1$ other transactions. Hence, differences are introduced in the database by the anonymization process. These differences can be considered as the information loss caused by the anonymization process.

Definition 2 (Information loss, *IL*)**.** The information loss, denoted as *IL*, is defined as the number of item differences between the original database D and the anonymized database D'. An item difference is an item that has been added, replaced or removed in a transaction.

For example, the IL of Table 2 with respect to Table 1 is 4 because there are four item differences between the two databases.

Problem Statement: Let there be a transaction database D, where each transaction T_j consists of a set of non-sensitive items (*QIDs*) and a set of sensitive items (*SI*). The goal of attaining k-anonymity is to obtain an anonymized transaction database where 1) transactions containing sensitive information are identical to at least $k-1$ other transactions, and; 2) the difference between the original database D and the anonymized database D' is as small as possible. The value k is called the anonymity degree, and is selected by users.

4 The Proposed PTA Anonymization System

To achieve the goal of k-anonymity for transactional data while minimizing information loss, we propose a system called PTA. The PTA system has three modules, which it applies one after the other to anonymize a dataset.

The first module is the **pre-processing module**. It treats all items in transactions as quasi-identifier attributes and encodes transactions as bitmaps. Then, it sorts the transactions using the Gray order to ensure that the most similar transactions appear consecutively in the database. This sorting order is used to facilitate the optimization of information loss, thereafter. A divide-and-conquer approach is then applied to group the sorted transactions into several segments. This step is done to reduce the execution time of the second module.

The second module is the **TSP module**. It applies a Traveling Salesman Problem (TSP) solving approach to each segment to find a cyclical loop between transactions (a local approximate solution). This process is applied to reduce the information loss in each segment.

The third module, named the **anonymization module** is applied after the second module. It groups similar transactions according to their similarity into groups. Then, a center point is calculated for each group using a mapping and majority-voting approach. The information loss that would be obtained by replacing all transactions by the center point in each group is then calculated. The group having the least information loss is then considered as an equivalence class. All transactions in that equivalence class are then replaced by its center point. The transactions in the equivalence class thus become identical and a minimal amount of information is lost. This process is repeated to create additional equivalence classes. Then, each transaction that has not been assigned to an equivalence class is then assigned to the most similar equivalence class in terms of the Hamming distance. This process increases anonymity for the database since the constraint of k-anonymity only requires that each transaction in an equivalence class is similar to at least (k-1) identical transactions. The flowchart of the designed PTA anonymization system is illustrated in Fig. 1.

4.1 The Pre-Processing Module

This module treats all items in a transaction database as QIDs and then encodes the database as bitmaps, where each transaction is a bitmap and each item is encoded using the Gray code [34]. Table 3 shows an example of how the decimal numbers from 0 to 7 are coded with the standard binary representation and with the Gray code.

TABLE 3: A gray coding example.

Decimal	0	1	2	3	4	5	6	7
Binary	000	001	010	011	100	101	110	111
Gray	000	001	011	010	110	111	101	100

Then, the bitmaps (transactions) are sorted by Gray code. The result is that similar transactions will appear consecutively, which is desirable to be able to minimize information loss. Note that this Gray sort is only used in the pre-processing phase since consecutive transactions may still be very different according to the Hamming distance measure, even after this sort.

To solve this limitation of the Gray sort for anonymizing transactional data, a divide-and-conquer mechanism is then used to partition the transactions into several segments. The number of segments can be set by the user according to his preferences. Then, for each segment, the TSP module will find the shortest cyclical loop, as it will be described in the next section. The result will be then used by the anonymization module. The divide-and-conquer mechanism is used to reduce the time complexity of the tasks performed by the TSP module. The complexity of the operations performed by the TSP module with or without the divide-and-conquer mechanism is analyzed below.

Complexity analysis: Assume that the original database D contains n transactions and that the database is divided into m segments. Thus, the size of each segment is n/m. For each segment, n/m cyclical loops are generated by the TSP module. The time complexity for finding each loop is $O((n/m)^2)$. Hence, the time complexity for each segment is $O((n/m)^3)$, and the time complexity for processing the whole original database is $O(m \times (n/m)^3) = O(n^3/m^2)$. If the divide-and-conquer mechanism is not used, the time complexity for processing the whole original database is $O(n^3)$. Therefore, when $m \geq 2$, the divide-and-conquer mechanism can the reduce time complexity and increase the efficiency of the designed approach.

Definition 3. Assume that the original database is denoted as D, and that the database encoded as bitmaps and sorted by the Gray order is denoted as $sortD$.

Definition 4. For the later anonymity process, we assume that $sortD$ is divided into m segments, each

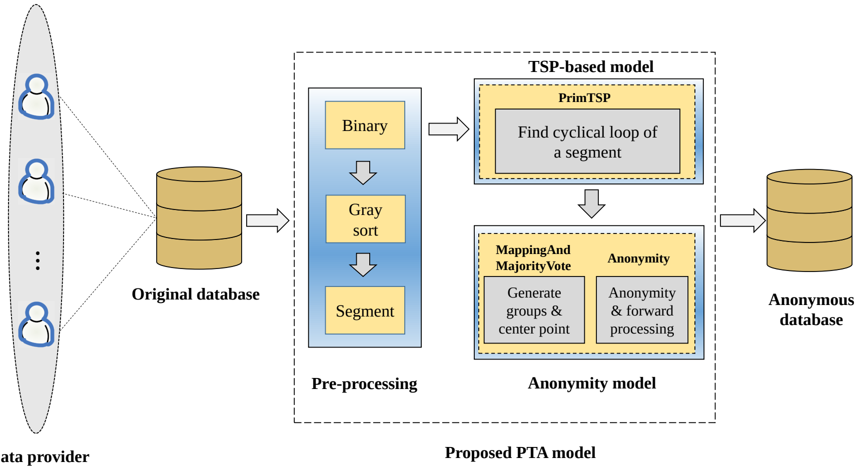


Fig. 1: Flowchart of the PTA system.

denoted as seg_l, where m is set by the user, and $1 \leq l \leq m$.

Besides the above operations, the pre-processing module also builds an adjacency matrix of transactions for each segment. This matrix is built using the Hamming distance as measure of similarity between transactions in each segment. The Hamming distance is defined as:

$$hd(T_{iQID}, T_{jQID}) = T_{iQID} \oplus T_{jQID}, \qquad (1)$$

where "$\oplus$" denotes the XOR operation, and T_{iQID}, T_{jQID} are respectively the QID part of T_i and T_j.

The adjacency matrix of each segment is used by the TSP module for finding the shortest cyclical path to minimize information loss in a segment.

4.2 The TSP Module

This module applies an approximation algorithm for the TSP problem to find a shortest path between transactions, for each segment seg_l ($1 \leq l \leq m$) and allows to minimize information loss in a segment since the distance between consecutive transactions (nodes) is minimized. The pseudo-code of the designed algorithm is shown in Algorithm 1.

Definition 5. Let the adjacency matrix based on the Hamming distance for a segment seg_l be denoted as $G_{seg_l}(V, E)$, where V is the set of transactions in the segment seg_l. Let the notation $hd(v_i, v_j)$ represents the length of the edge $(v_i \rightarrow v_j) \in E$ between two transactions v_i and v_j according to the Hamming distance.

The designed PrimTSP algorithm (Algorithm 1) takes as input a set of segments, and outputs an optimized cyclical loop for each segment. The algorithm processes each segment separately to find its shortest path using local optimization to minimize information loss (Lines 4 to 27). Each transaction is represented as a vector (node) in $G_{seg_l}(V, E)$ and the Hamming distance between vectors is considered as the amount of information loss. For each row in $G_{seg_l}(V, E)$, the first transaction of the row is both set as the start (line 8) and destination (Line 9) nodes for finding its shortest cyclical loop using a TSP approach (Lines 12 to 24). Thus, an optimized cyclical path that minimizes information loss for each row (transaction) is calculated. The nodes having the least Hamming distance to either the start node or destination node are then found and will be considered as the next

Algorithm 1: PrimTSP Algorithm

```
   Input: $seg\_set$, the set of segments.
   Output: $OptLoop\_set$, a set containing an optimized cyclical loop for
           each segment.
1  for each $seg_l \in seg\_set$, $1 \leq l \leq m$ do
2      calculate the adjacency matrix of $G_{seg_l}(V, E)$ using the
         Hamming distance;
3      set $GlOptLoop.IL := \infty$;
4      for each node $v_i$ in $G_{seg_l}$ do
5          $LocOptloop(v_i) \leftarrow \emptyset$;
6          set $LocOptloop(v_i).IL := 0$;
7          set $NVistN\_set \leftarrow V$;
8          $Star \leftarrow v_i$;
9          $Dest \leftarrow v_i$;
10         $LocOptloop(v_i) \leftarrow v_i$;
11         remove($v_i, NVistN\_set$);
12         while $NVistN\_set \neq \emptyset$ do
13             $v_{Near\_star} = argmin(\{v_j | hd(Star, v_j), v_j \in$
                 $NVistN\_set\})$;
14             $v_{Near\_dest} = argmin(\{v_n | hd(Dest, v_n), v_n \in$
                 $NVistN\_set, n \neq j\})$;
15             if $hd(Star, v_{Near\_star}) > hd(Dest, v_{Near\_dest})$
                 then
16                 $LocOptloop(v_i) \leftarrow v_{Near\_dest}$;
17                 $Dest \leftarrow v_{Near\_dest}$;
18                 $LocOptloop(v_i).IL += hd(Dest, v_{Near\_dest})$;
19                 remove ($v_{Near\_dest}, NVistN\_set$);
20             else
21                 $LocOptloop(v_i) \leftarrow v_{Near\_star}$;
22                 $Star \leftarrow v_{Near\_star}$;
23                 $LocOptloop(v_i).IL += hd(Star, v_{Near\_star})$;
24                 remove ($v_{Near\_star}, NVistN\_set$);
25         if $LocOptloop(v_i).IL < GlOptLoop.IL$ then
26             $GlOptLoop.IL := LocOptloop(v_i).IL$;
27             $GlOptLoop \leftarrow LocOptloop(v_i)$;
28     $OptLoop\_set \leftarrow GlOptLoop$;
29 return $OptLoop\_set$;
```

nodes in this path (Lines 15 to 24). This process is then repeated until all nodes (transactions) have been visited. When the algorithm terminates, an optimized cyclical path has been found for each segment, to minimize the information loss. The set of loops found for all segments can be considered as a global solution that minimize the information loss (Lines 28 to 29).

4.3 The Anonymization Module

In previous works, *k*-anonymity [17] has been obtained by first finding the center point of each group (segment) and then replacing each group member (transaction) by that center point. However, performing this in the context of this paper is not a trivial task since information loss must be taken into account, and the choice of a center point influences the amount of information lost.

The proposed solution to this problem is the following. By applying the TSP module, an optimized shortest path is obtained for each segment, which minimizes information loss. Each vector (node) in the shortest path is then mapped to several groups using a symmetric mapping mechanism, which consists of mapping consecutive transactions into the same group for later anonymization. A majority-voting mechanism is then used to find the center point of the group to perform anonymization. After that, the sum of the Hamming distances between all vectors to the center point is calculated for each group to obtain its total information loss. The detailed procedure for mapping and majority-voting are described in Algorithm 2.

Algorithm 2: MapAndMajorityVote

```
Input: $OptLoop$, the optimal cyclical loop of a segment $seg_l$; $k$, the
       degree of anonymity.
Output: $MG$, the candidate group set of a segment $seg_l$.
1  $MG \leftarrow \emptyset$;
2  for each node $v_i \in OptLoop$ do
3      create a group $G_i$;
4      set $range = (k-1)/2 + (k-1)\%2$;
5      for $j = i - (range - 1)$ to $i + (range - 1)$ do
6          $G_i.tidlist \leftarrow v_{(j+|OptLoop|)\%|OptLoop|}.tid$;
7      if $k\%2 == 1$ then
8          $G_i.tidlist \leftarrow v_{(i-range+|OptLoop|)\%|OptLoop|}.tid$;
9          $G_i.tidlist \leftarrow v_{(i+range+|OptLoop|)\%|OptLoop|}.tid$;
10     else
11         if $hd(v_{(i-range+|OptLoop|)\%|OptLoop|}.tid, v_i.tid) <$
             $hd(v_{(i+range+|OptLoop|)\%|OptLoop|}.tid, v_i.tid)$ then
12             $G_i.tidlist \leftarrow v_{(i-range+|OptLoop|)\%|OptLoop|}.tid$;
13         else
14             $G_i.tidlist \leftarrow v_{(i+range+|OptLoop|)\%|OptLoop|}.tid$;
15     $MG \leftarrow G_i$;
16 for each $G_i \in MG$ do
17     for each $T_p \in G_i$ do
18         $G_i.data+ = T_p$;
19     for j=0 to $G_i.data.length$ do
20         if $G_i.data[j] > k/2$ then
21             $G_i.data[j] = 1$;
22         else
23             $G_i.data[j] = 0$;
24     for each $T_p \in G_i$ do
25         $G_i.IL+ = hd(T_p, G_i.data)$;
26 return $MG$;
```

Definition 6 (Group)**.** The structure of each group G_i in a segment contains three fields: $1)$.**tidlist:** the IDs of transactions contained in G_i; $2)$.**data:** the center point of the group; $3)$.**IL:** the information loss of G_i.

In Algorithm 2, the range of the processed vector (transactions) in a shortest path is used to find the most similar transactions and group them together based on the predefined anonymity degree *k* (Lines 2 to 15). Consecutive transactions in a shortest path are then selected and mapped together as a group. This symmetric mapping procedure ensures that the grouped transactions produce a minimal loss of information. A majority-voting procedure is then applied (Lines 17 to 23) to find the center point of the mapped group and calculate its total information loss (Lines 24 to 25). The result is a set of groups for each segment and a center point for each group.

After the mapping and majority-voting procedure have been applied, the mapped group with the least information loss is then used to create an equivalence class. The transactions in that equivalence class are replaced by the center point. The other groups are then processed in ascending order of information loss to create additional equivalence classes. By this process of building equivalence classes, all transactions in an equivalence class have at least $(k-1)$ identical transactions.

Then, each transaction that has not been assigned to any equivalence class is compared with the center points of all equivalence classes. The transaction is replaced by the center point providing the least information loss. This process increases the anonymity degree of that equivalence class to $(k+m)$, where $m(< k)$ unassigned transactions are added to it. Details of the anonymization algorithm are given in Algorithm 3.

The mapped groups are first discovered (Line 3) and then the mapped groups are used to find the equivalence classes to perform anonymization. The groups are processed by ascending order of information loss. A group is chosen to form an equivalence class if no transactions in this group are members of other groups according to the intersection operation (Lines 6 to 12). Thereafter, each transactions that has not yet been assigned to any equivalence class is assigned to the closest equivalence class (Lines 13 to 15). This forward procedure can ensure that each transaction in the anonymized database satisfies *k*-anonymity and cannot be re-identified. Each transaction in an equivalence class are then replaced by its center point of it (Lines 16 to 19) and the information loss is calculated (Line 20). It is thus guaranteed that at least *k*-anonymity is achieved.

Algorithm 3: Anonymization

```
Input: $OptLoop\_set$, a set of optimal loops for $m$ segments; $k$, the
       degree of anonymity.
Output: $D'$, an anonymized database.
set $D' \leftarrow \emptyset$;
for each $OptLoop \in OptLoop\_set$ do
    $MG \leftarrow MapAndMajorityVote(OptLoop)$;
    set $EquiC\_set \leftarrow \emptyset$;
    set $EquiC\_set.IL := 0$;
    while $MG \neq \emptyset$ do
        $EquiC\_set \leftarrow \{G_i | argmin\{G_i.IL\}, G_i \in MG\}$
          $EquiC\_set.IL += G_i.IL$;
        remove($G_i, MG$);
        for each $G_p$ in $MG$ do
            if $G_i.tidlist \cap G_p.tidlist \neq \emptyset$ then
                remove($G_p, MG$);
    for each $T_q \in OptLoop \bigwedge T_q \notin \forall EquiC \in EquiC\_set$ do
        $\{EquiC | argmin\{hd(EquiC.data, T_q)\}, EquiC \in$
          $MG\} \leftarrow T_q$;
        $EquiC.IL += hd(EquiC.data, T_q)$;
    for each $EquiC \in EquiC\_set$ do
        for each $T_i \in EquiC.tidlist$ do
            $T_i.data = EquiC.data$;
            $D' \leftarrow T_i$;
        $D'.IL += EquiC.IL$;
return $D'$;
```

5 An Illustrated Example

An example is now provided to illustrate how the algorithm is applied, step by step. In this example, suppose that the anonymity degree k is set to 3 and that the number of segments is initially set to 2 (these parameters can be adjusted by users according to their preferences). Consider the database depicted in Fig. 2(a) containing thirteen transactions. This database is first transformed into quasi-identifiers and each attribute is represented as a binary value. The resulting database is shown in Fig. 2(b). Then, transformed transactions are sorted by the Gray order. The sorted database is presented in Fig. 2(c).

QID	*SI*
B, D	S_1
A, B, C	S_2
C, D, E	S_3
A, C, D, E	S_3
A, B, D	S_2
B, D, E	S_4
A, C	S_5
D, E	S_6
A, E	S_7
B, D	S_2
B, D	S_1
A, C, E	S_5
B, C, D	S_7

Binary

TID	QID					*SI*
	A	*B*	*C*	*D*	*E*	
T_1	0	1	0	1	0	S_1
T_2	1	1	1	0	0	S_2
T_3	0	0	1	1	1	S_3
T_4	1	0	1	1	1	S_3
T_5	1	1	0	1	0	S_2
T_6	0	1	0	1	1	S_4
T_7	1	0	1	0	0	S_5
T_8	0	0	0	1	1	S_6
T_9	1	0	0	0	1	S_7
T_{10}	0	1	0	1	0	S_2
T_{11}	0	1	0	1	0	S_1
T_{12}	1	0	1	0	1	S_5
T_{13}	0	1	1	1	0	S_7

Gray sort

TID	QID				
	A	*B*	*C*	*D*	*E*
T_8	0	0	0	1	1
T_3	0	0	1	1	1
T_{13}	0	1	1	1	0
T_1	0	1	0	1	0
T_{10}	0	1	0	1	0
T_{11}	0	1	0	1	0
T_6	0	1	0	1	1
T_5	1	1	0	1	0
T_2	1	1	1	0	0
T_7	1	0	1	0	0
T_{12}	1	0	1	0	1
T_4	1	0	1	1	1
T_9	1	0	0	0	1

Fig. 2: Illustrations of the transformation and sorting process.

The divide-and-conquer mechanism is then applied to the sorted database to divide it into segments. The resulting segments are shown in Fig. 3. Segmentation is performed so that each segment can be independently processed for anonymization. In this section, the first segment is considered to illustrate the anonymization process.

Segment	TID	QID					*SI*
		A	*B*	*C*	*D*	*E*	
seg_1	T_8	0	0	0	1	1	S_6
	T_3	0	0	1	1	1	S_3
	T_{13}	0	1	1	1	0	S_7
	T_1	0	1	0	1	0	S_1
	T_{10}	0	1	0	1	0	S_2
	T_{11}	0	1	0	1	0	S_1
	T_6	0	1	0	1	1	S_4

Segment	TID	QID					*SI*
		A	*B*	*C*	*D*	*E*	
seg_2	T_5	1	1	0	1	0	S_2
	T_2	1	1	1	0	0	S_2
	T_7	1	0	1	0	0	S_5
	T_{12}	1	0	1	0	1	S_5
	T_4	1	0	1	1	1	S_3
	T_9	1	0	0	0	1	S_7

Fig. 3: The two segments created from the sorted database.

For the first segment, the adjacency matrix based on the Hamming distance (seg_1) is built. It is depicted in Fig. 4.

	T_8	T_3	T_{13}	T_1	T_{10}	T_{11}	T_6
T_8	∞	1	3	2	2	2	1
T_3	1	∞	2	3	3	3	2
T_{13}	3	2	∞	1	1	1	2
T_1	2	3	1	∞	0	0	1
T_{10}	2	3	1	0	∞	0	1
T_{11}	2	3	1	0	0	∞	1
T_6	1	2	2	1	1	1	∞

Fig. 4: The built matrix of Hamming distance of segment 1.

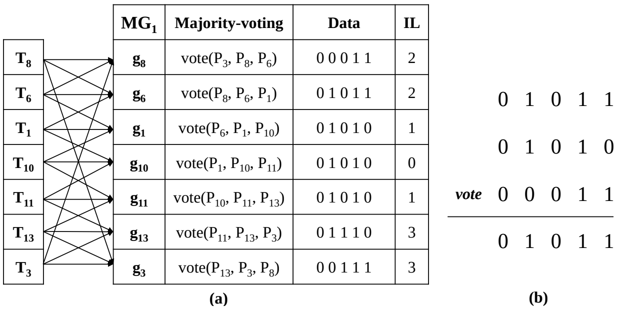
Each transaction in the first segment is then processed to find its shortest cyclical path using the PrimTSP algorithm. This latter first sets the start and destination node of the cyclical loop to transaction T_8. Recall that the distance between two nodes is considered as the information loss for the purpose of anonymization. From Fig. 4, it can be found that T_3 and T_6 have the smallest information loss with respect to T_8 and that $hd(T_8, T_3) = 1$, $hd(T_8, T_6) = 1$. Transaction T_3 is first considered as the node preceding the start node T_8, and hence the start node is set to T_3. The node that is the closest to the start node is T_{13}. T_6 is still the closest node to the destination node T_8 and $hd(T_8, T_6) = 1 < hd(T_3, T_{13}) = 2$. Thus, T_6 is considered as the node succeeding the destination node T_8. The destination node is then changed to T_6. This process is repeated for the other unvisited nodes. When that process ends, the shortest cyclical loop of T_8 has been obtained. This loop is shown in Fig. 5, where numbers above arrows indicate the order used for selecting nodes.

$$T_8 \xrightarrow{①} T_3 \xrightarrow{⑦} T_{13} \xleftarrow{⑥} T_{11} \xleftarrow{⑤} T_{10} \xleftarrow{④} T_1 \xleftarrow{③} T_6 \xleftarrow{②} T_8$$

Fig. 5: The cyclical loop of transaction T_8 in the first segment.

The total loss of information for a loop is calculated as the sum of the distances between all nodes in the loop. In this example, the distances between nodes are calculated as $hd(T_8, T_3) = 1$; $hd(T_3, T_{13}) = 2$; $hd(T_{13}, T_{11}) = 1$; $hd(T_{11}, T_{10}) = 0$; $hd(T_{10}, T_1) = 0$; $hd(T_1, T_6) = 1$; $hd(T_6, T_8) = 1$. Hence, the total information loss for this loop is 1 + 2 + 1 + 0 + 0 + 1 + 1 (= 6). Each other segment is processed in the same way to find its shortest cyclical path and calculate its information loss. After that, the algorithm finds the cyclical path having the least information loss according to the matrix. This path will then be used by the mapping and majority-voting approach. In the first segment, the cyclical shortest path is (T_8, T_6, T_1, T_{10}, T_{11}, T_{13}, T_3).

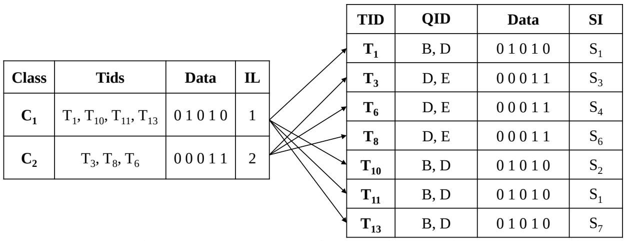
After that, transactions in the cyclical path are mapped to several groups. The center point and information loss of each group is calculated. The process of majority-voting is illustrated in Fig. 6(b) for three transactions 01011, 01010, and 00011. The majority-voting procedure selects the most frequent binary value for each position (attribute). For example, there are two 1 and one 0 in the second position in Fig. 6(b). Thus, the value 1 holds the majority and is chosen for the second position. In this example, 0, 1, 0, 1 and 1 are respectively the most frequent values for each of the five positions. Thus, the result of the majority-voting procedure for 01011, 01010 and 00011 is 01011. The result of the mapping and majority-voting procedure for the running example is shown in Fig. 6(a).



MG1	Majority-voting	Data	IL
g_8	vote(P_3, P_8, P_6)	0 0 0 1 1	2
g_6	vote(P_8, P_6, P_1)	0 1 0 1 1	2
g_1	vote(P_6, P_1, P_{10})	0 1 0 1 0	1
g_{10}	vote(P_1, P_{10}, P_{11})	0 1 0 1 0	0
g_{11}	vote(P_{10}, P_{11}, P_{13})	0 1 0 1 0	1
g_{13}	vote(P_{11}, P_{13}, P_3)	0 1 1 1 0	3
g_3	vote(P_{13}, P_3, P_8)	0 0 1 1 1	3

Fig. 6: The mapped groups, their center points, and their information loss.

Then, groups are processed one-by-one by ascending order of information loss. In this example, the group g_{10} is first processed since its information loss is 0. It is used to create the first equivalence class C_1. The groups g_1 and g_{11} are then processed. But it is found that there is some overlap as transactions T_1 and T_{10} are in g_1, and transactions T_{10} and T_{11} are in g_{11}. Thus, these groups are not used to create equivalence classes. Then, the other groups are considered to find a group that has no transaction overlap with other groups. In this example, the group g_8 meet this criterion and is thus used to create the second equivalence class C_2. Thus, two equivalence classes are obtained, that is C_1 = {T_1, T_{10}, T_{11}} and C_2 = {T_3, T_8, T_6}. However, in segment 1, there is still a transaction (T_{13}) that has not been assigned to any equivalence class. Thus, T_{13} is compared with the center point of each equivalence classe and it is assigned to the equivalence class having the closest center point (that would produce the smallest information loss). Hence, T_{13} is assigned to class C_1. Transactions in each equivalence class are then replaced by the data contained in its center point. As a result, transactions in each equivalence class become identical. The result for segment 1 is shown in Fig. 7.



Class	Tids	Data	IL
C_1	T_1, T_{10}, T_{11}, T_{13}	0 1 0 1 0	1
C_2	T_3, T_8, T_6	0 0 0 1 1	2

TID	QID	Data	SI
T_1	B, D	0 1 0 1 0	S_1
T_3	D, E	0 0 0 1 1	S_3
T_6	D, E	0 0 0 1 1	S_4
T_8	D, E	0 0 0 1 1	S_6
T_{10}	B, D	0 1 0 1 0	S_2
T_{11}	B, D	0 1 0 1 0	S_1
T_{13}	B, D	0 1 0 1 0	S_7

Fig. 7: The anonymized database (segment 1).

Segment 2 is then processed in the same way. Anonymity is achieved, and in particular for the transactions {T_1, T_{10}, T_{11}, T_{13}}, 4-anonymity is obtained.

6 Experimental Evaluation

To verify the effectiveness and efficiency of the proposed PTA system, substantial experiments have been conducted on five real-world datasets, namely chess [35], mushroom [35], pumsb [35], connect [35], and accidents [35], and a synthetic named T10I4D100K [36], which was generated using the IBM Quest Synthetic Data Generator. All the algorithms were implemented in Java and experiments were carried out on a personal computer equipped with an Intel Core i3-4160 dual-core processor and 4 GB of RAM, running the 32-bit Microsoft Windows 7 operating system. The parameters and the characteristics of the used datasets are shown in [35].

The experiments compare the designed approach with the state-of-the-art Gray-TSP algorithm [18] and GSC algorithm [17]. The three algorithms are compared in terms of information loss and runtime for various number of segments ***Segs*** and various ***k*** values.

6.1 Evaluation of the Divide-and-Conquer Mechanism

This section assesses the efficiency of the divide-and-conquer mechanism, proposed to reduce the runtime of the anonymization process. Results for different k values and number of segments are shown in Fig. 8.

In Fig. 8, it can be observed that the runtime of the proposed algorithm is greatly influenced by the number of segments, for all values of k. When the number of segments is increased, runtime decreases. The reason is that the divide-and-conquer mechanism

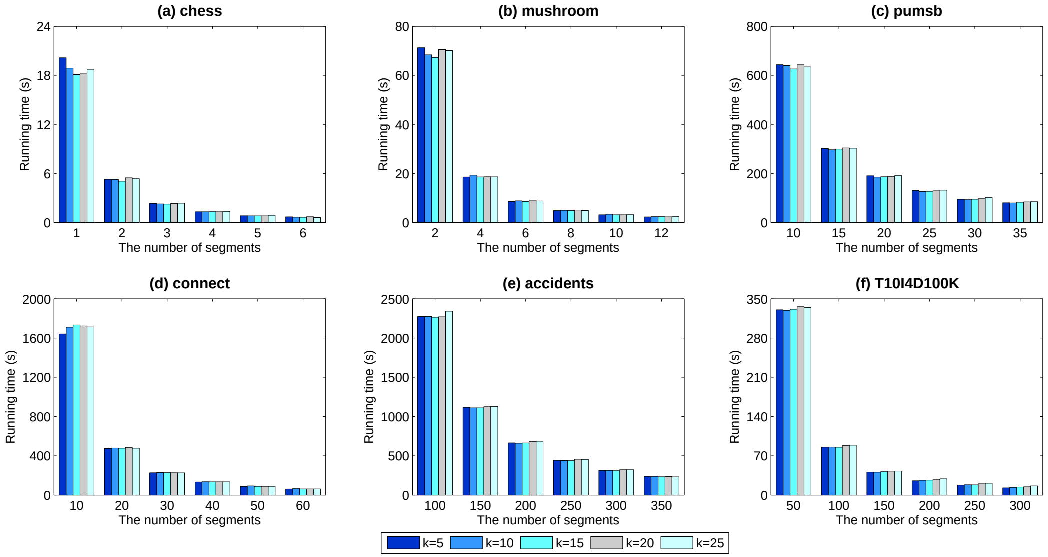


Fig. 8: Runtimes for various number of segments and various k values.

divides the data into segments of smaller size that can be anonymized independently. For example, in Fig. 8(c), the runtime is about 600 seconds for 10 segments, and it decreases to 300 seconds when the number of segments is increased to 15. It can also be observed in Fig. 8(f) that the runtime for 50 segments is about 330 seconds, and that it decreases to 80 seconds when the number of segments is raised to 100. Thus, the designed divide-and-conquer mechanism can considerably increase the performance of the proposed anonymization approach. In addition, it can be observed that runtime is not influenced by the value of k. The reason is that the time spent by the anonymization module is small compared to the time required for finding a cyclical loop for each segment. However, increasing k influences the anonymity degree and the information loss. The results about information loss for various number of segments are shown in Fig. 9, where information loss is expressed as a ratio (the amount of information that has been lost divided by the total amount of information).

In Fig. 9, it can be observed that the number of segments has not much influence on information loss. When the number of segments is increased, information loss remains more or less the same. The reason is that the divide-and-conquer approach is only used to partition the transactions for the later anonymization process. It is designed to speed up the computation by dividing the data into segments that can then be anonymized independently. The ratio of information loss rapidly increases when k is increased. This is reasonable since when k is increased, there are more transactions in each segment or equivalence class, and all transactions in an equivalence class become identical to attain k anonymity. Thus, the ratio of information loss increases.

6.2 Information Loss

This section compares the information loss of the proposed PTA system with the state-of-the-art Gray-TSP and GSC algorithms. The parameter k is set to 15, as it is the median value used in previous experiments, and the number of segments is varied. The results are shown in Fig. 10.

In Fig. 10, it can be observed that the designed PTA system performs well on the six datasets. The state-of-the-art GSC algorithm does not apply a divide-and-conquer mechanism for anonymization. Thus, the information loss remains the same when the number of segments is varied. The ratio of information loss for the proposed PTA system and Gray-TSP algorithm generally decreases when the number of segments is increased. The reason is that when transactions are divided into several segments, transactions in each segment are highly similar. Thus, fewer transformations are needed to modify the transactions in each segment or equivalence class to obtain k-anonymity. For example, the information loss for the designed algorithm in Fig. 10(b) and 100 segments is 19.5%, while Gray-TSP obtains about 28%. When the number of segments is increased to 200, the information loss of the proposed PTA system is 19% and the information loss of Gray-TSP decreases to 23%. The GSC algorithm has clearly the worst performance among the three compared algorithms, as it can be observed in Fig. 10(c), Fig. 10(e), and Fig. 10(f). Overall, the proposed PTA system reduces the loss of information for anonymizing datasets. A second experiment was performed to compare information loss for a fixed number of segments, while varying the parameter k. Results for the six datasets are shown in Fig. 11. In this experiment, the number of segments used for the six datasets was respectively 60, 150, 800, 1,250, 6,000, and 1,000 (the median numbers of segments used in

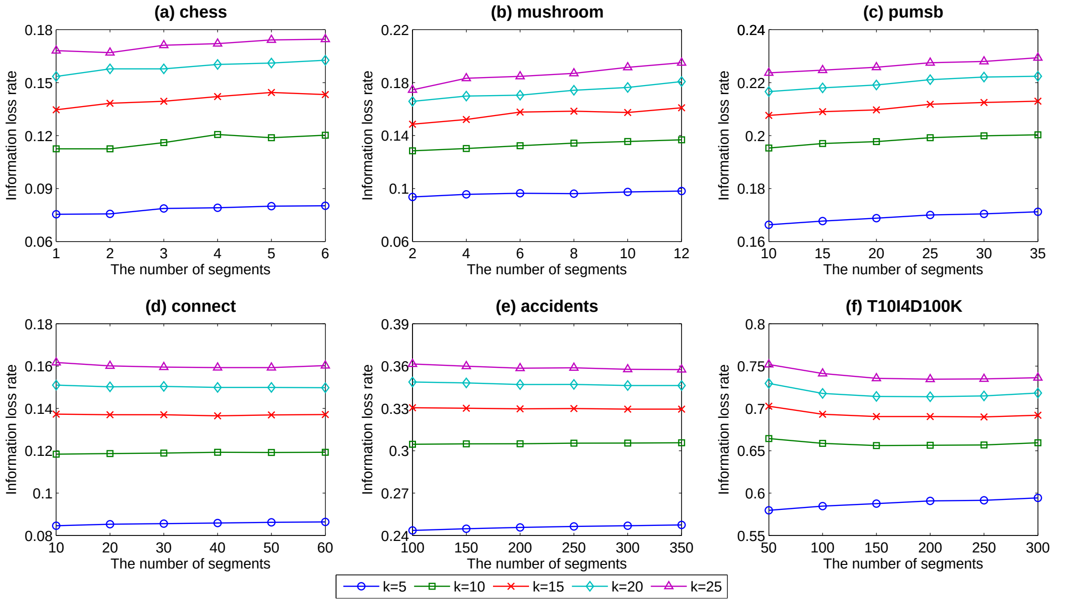


Fig. 9: Information loss for various number of segments.

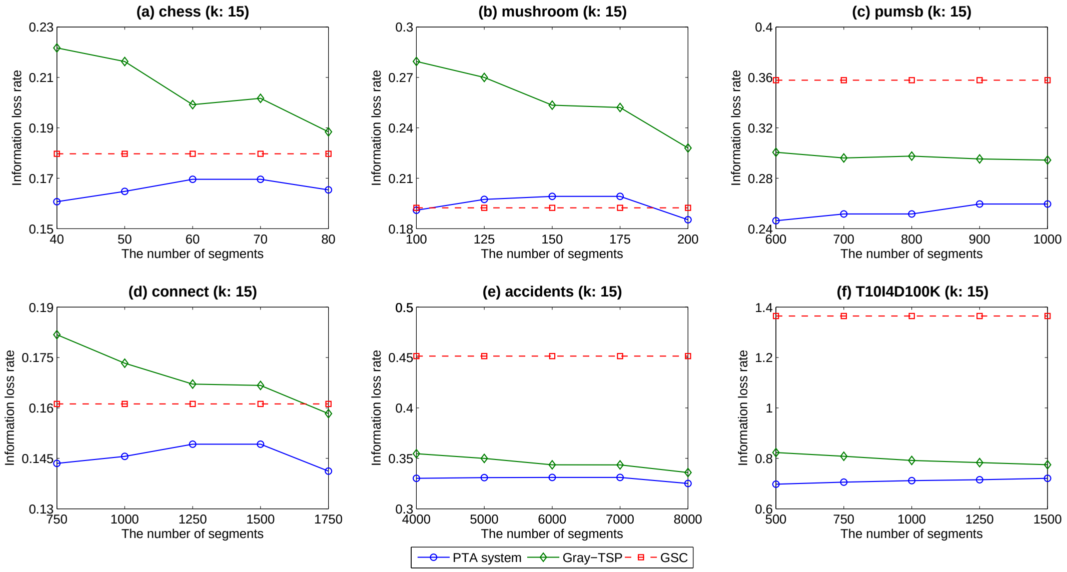


Fig. 10: Information loss for a fixed k and various number of segments.

Fig. 10).

It can be observed in Fig. 11 that the proposed PTA system globally has better results than the other two algorithms when k is varied. As k is increased, the ratio of information loss decreases. This is reasonable since when k is increased, more transactions need to be anonymized and transformed into identical transactions for each segment/equivalence class. It can also be seen that the proposed PTA system outperforms the other two algorithms, and the GSC algorithm generally has the worse performance, such as in Fig. 10(c), Fig. 10(e), and Fig. 10(f). Overall, the value of k influences the ratio of information loss, while the number of segments does not have a strong influence on information loss.

6.3 Runtime

In this section, the runtimes of the three algorithms are first compared for a fixed value of k, while varying the number of segments, on the same datasets. Results are shown in Fig. 12.

It can be observed in Fig. 12 that the GSC algorithm performs well on all datasets except for the accidents dataset (results shown in Fig. 12(f)). Although the GSC algorithm is sometimes faster than the proposed PTA system, their runtimes are very similar, when the number of segments is varied. The proposed PTA system and the GSC algorithms are both about one to two orders magnitude faster than the Gray-TSP algorithm. For example, in Fig. 12(a), the proposed PTA system requires nearly 10^{-2} seconds while Gray-TSP spends more than 10^{1} seconds for anonymization.

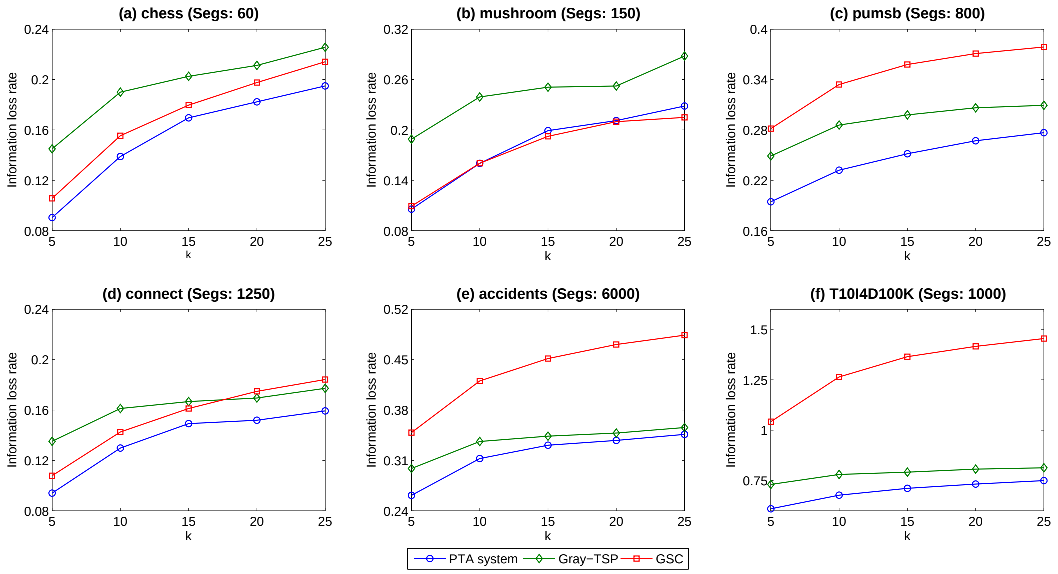


Fig. 11: Information loss for a fixed number of segments and various values of k.

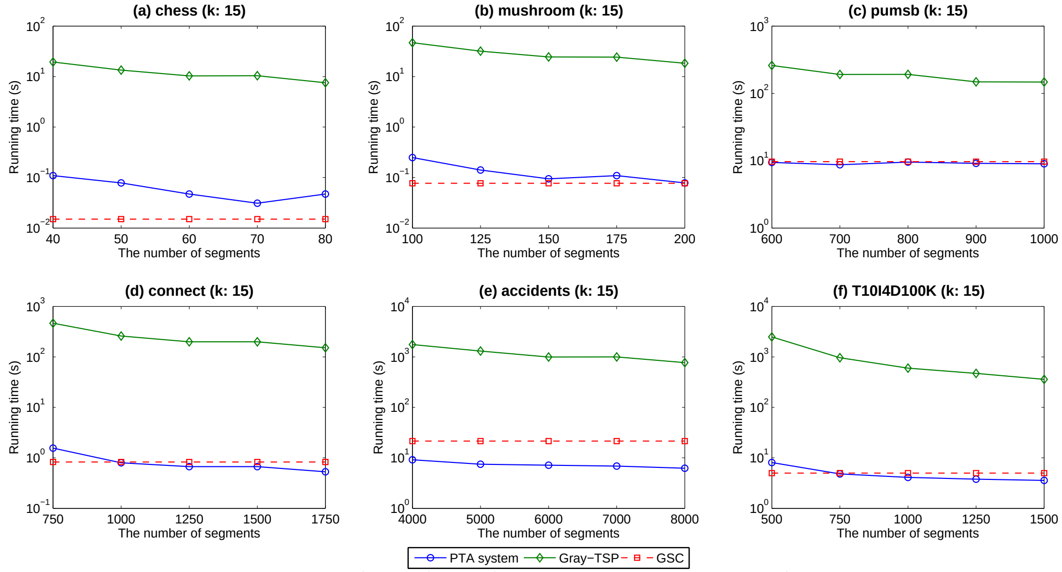


Fig. 12: Runtime for a fixed k and various number of segments.

The reason is that the Gray-TSP algorithm utilizes a genetic algorithm to calculate the shortest TSP path to anonymize transactions, and the evolutionary process requires a huge amount of time. Results in terms of runtime for a fixed number of segments and various k values are shown in Fig. 13.

In general, the proposed PTA system outperforms the Gray-TSP algorithm. But for the chess, mushroom, and pumsb datasets, the proposed PTA system has nearly the same runtime as the GSC algorithm. However, the runtime difference between these algorithms is not huge and the information loss of the GSC algorithm is much greater than the designed system, as it can be observed in Fig. 10. For example, in Fig. 13(b), the runtimes of GSC and the designed system are both less than 10^{-1} seconds, while Gray-TSP exceeds 10^1 seconds. Besides, the proposed PTA system and the GSC algorithm are one or two orders magnitude faster than the Gray-TSP algorithm, when k is varied. When k is increased, the runtimes of Gray-TSP and GSC are steady while the runtime of the proposed PTA system is slightly influenced by the value of k.

Overall, it can be concluded that the Gray-TSP algorithm takes more time than the two other algorithms since it uses the evolutionary process of genetic algorithms to find similar transactions for anonymization. The GSC algorithm and the proposed PTA system have similar runtimes but the GSC algorithm produces a greater loss of information compared to the designed system. In summary, the proposed PTA system can thus achieve better performance than the other two algorithms and the divide-and-conquer mechanism can efficiently reduce the runtime for

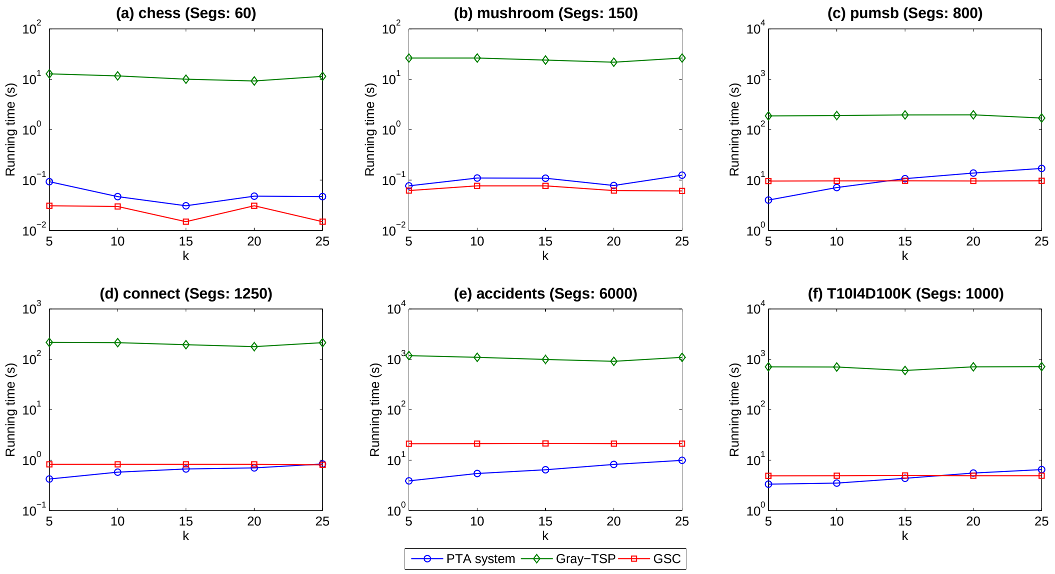


Fig. 13: Runtime for a fixed number of segments and various values of k.

attaining anonymity.

7 Conclusion

Numerous algorithms have been designed to anonymize relational data. But it is also a critical issue to prevent the disclosure of sensitive information in transactional data. Works that have been designed for transactional data typically produce a high information loss. In this paper, we have presented a novel anonymization system called PTA, which consists of three modules. It anonymizes transactional data with a small loss of information loss and it is also very fast compared to the state-of-the-art algorithms for anonymizing transactional data. Substantial experiments have been carried to compare the performance of the designed system to the state-of-the-art algorithms for anonymizing transactional data in terms of runtime and information loss.

References

[1] M. S. Chen, J. Han and P. S. Yu, “Data mining: An overview from a database perspective,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 866–883, 1996.

[2] U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, “From data mining to knowledge discovery in databases,” *AI magazine*, vol. 17, no. 3, p. 37, 1999.

[3] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[4] J. Bobadilla, F. Ortega, A. Hernando and A. Gutierrez, “Recommender systems survey,” *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013.

[5] G. Jeh and J. Widom, “Scaling personalized web search,” *International Conference on World Wide Web*, pp. 271–279, 2003.

[6] L. Shou, H. Bai, K. Chen and G. Chen, “Supporting privacy protection in personalized web search,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 453–467, 2014.

[7] L. Sweeney, “Achieving *k*-anonymity privacy protection using generalization and suppression,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 571–588, 2002.

[8] A. Machanavajjhala, D. Kifer and J. Gehrke, “*l*-diversity: Privacy beyond *k*-anonymity,” *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, pp. 1–12, 2007.

[9] P. Samarati and L. Sweeney, “Generalizing data to provide anonymity when disclosing information,” *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, p. 188, 1998.

[10] P. Samarati, “Protecting respondents identities in microdata release,” *IEEE Transaction on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[11] L. Sweeney, “*K*-anonymity: a model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[12] R. J. Bayardo and R. Agrawal, “Data privacy through optimal *k*-anonymization,” *The International Conference on Data Engineering*, pp. 217–228, 2005.

[13] K. LeFevre, D. J. DeWitt and R. Ramakrishnan, “Incognito: Efficient full-domain *k*-anonymity,” *ACM SIGMOD International Conference on Management of Data*, pp. 49–60, 2005.

[14] J. Xu, W. Wang and J. Pei, "Utility-based anonymization using local recoding," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–790, 2006.

[15] G. Ghinita, Y. Tao and P. Kalnis, "On the anonymization of sparse high-dimensional data," *The International Conference on Data Engineering*, pp. 715–724, 2008.

[16] M. Terrovitis, N. Mamoulis and P. Kalnis, "Privacy-preserving anonymization of set-valued data," *Proceedings of the Very Large Data Bases Endowment*, vol. 1, no. 1, pp. 115–125, 2008.

[17] S. L. Wang, Y. C. Tsai, H. Y. Kao and T. P. Hong, "On anonymizing transactions with sensitive items," *Applied Intelligence*, vol. 41, no. 4, pp. 1043–1058, 2014.

[18] M. Xue, P. Karras and C. Rassi, "Anonymizing set-valued data by non-reciprocal recoding," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1050–1058, 2012.

[19] S. Hajian, J. Domingo-Ferrer and O. Farrs, "Generalization-based privacy preservation and discrimination prevention in data publishing and mining," *Data Mining and Knowledge Discovery*, vol. 28, no. 5-6, pp. 1158–1188, 2014.

[20] 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, 2012.

[21] M. Z. Islam and L. Brankovic, "Privacy preserving data mining: A noise addition framework using a novel clustering technique," *Knowledge-Based Systems*, vol. 24, no. 8, pp. 1214–1223, 2011.

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

[23] C. W. Lin, B. Zhang, K. T. Yang and T. P. Hong, "Efficiently hiding sensitive itemsets with transaction deletion based on genetic algorithms," *The Scientific World Journal*, vol. 2014, p. 13, 2014.

[24] S. Kisilevich, L. Rokach, Y. Elovici and B. Shapira, "Efficient multidimensional suppression for *k*-anonymity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 3, pp. 334–347, 2010.

[25] S. L. Wang, Y. C. Tsai, H. Y. Kao and T. P. Hong, "Extending suppression for anonymization on set-valued data," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 12, pp. 6849–6863, 2011.

[26] O. Abul, F. Bonchi and M. Nanni, "Anonymization of moving objects databases by clustering and perturbation," *Information Systems*, vol. 35, no. 8, pp. 884–910, 2010.

[27] G. Poulis, G. Loukides, A. Gkoulalas-Divanis and S. Skiadopoulos, "Anonymizing data with relational and transaction attributes," *Machine Learning and Knowledge Discovery in Databases*, pp. 353–369, 2013.

[28] K. Doka, M. Xue, D. Tsoumakos and P. Karras, "*k*-Anonymization by freeform generalization," *ACM Symposium on Information, Computer and Communications Security*, pp. 519–5303, 2015.

[29] Y. Wang, L. Xie, B. Zheng and K. C. K. Lee, "High utility *k*-anonymization for social network publishing," *Knowledge and Information Systems*, vol. 41, no. 3, pp. 697–725, 2014.

[30] S. Chettri and B. Borah, "Anonymizing classification data for preserving privacy," *Security in Computing and Communications*, pp. 99–109, 2015.

[31] Y. Xu, K. Wang, A. W. C. Fu and P. S. Yu, "Anonymizing Transaction Databases for Publication," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 767–775, 2008.

[32] G. Ghinita, P. Kalnis and Y. Tao, "Anonymous publication of sensitive transactional data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 2, pp. 161–174, 2011.

[33] C. H. Hsu and H. P. Tsai, "KAMP: Preserving *k*-anonymity for combinations of patterns," *The International Conference on Mobile Data Management*, pp. 97–102, 2013.

[34] F. Gray, "Pulse code communication," *U.S. Patent 2632058*, 1953-3-17.

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

[36] R. Agrawal and R. Srikant, "Quest synthetic data generator," *IBM Almaden Research Center*, http://www.Almaden.ibm.com/cs/quest/syndata.html.