applied sciences

MDPI

Article

A Graph-Based Differentially Private Algorithm for Mining Frequent Sequential Patterns

Miguel Nunez-del-Prado [1,2], Yoshitomi Maehara-Aliaga [1], Julián Salas [3,4,*], Hugo Alatrista-Salas [5] and David Megías [4,6,*]

[1] Peru Research Development, and Innovation (PERU IDI), Lima 15047, Peru; miguel.nunez@peruidi.com (M.N.-d.-P.); yoshitomimaehara@gmail.com (Y.M.-A.)
[2] Instituto de Investigación, Universidad Andina del Cusco, Cusco 08006, Peru
[3] Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili (URV), 43007 Tarragona, Spain
[4] Center for Cybersecurity Research of Catalonia (CYBERCAT), 08860 Barcelona, Spain
[5] Science and Engineering School, Pontificia Universidad Católica del Perú (PUCP), Lima 5088, Peru; halatrista@pucp.pe
[6] Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC), 08860 Barcelona, Spain
* Correspondence: julian.salas@urv.cat (J.S.); dmegias@uoc.edu (D.M.)

Abstract: Currently, individuals leave a digital trace of their activities when they use their smartphones, social media, mobile apps, credit card payments, Internet surfing profile, etc. These digital activities hide intrinsic usage patterns, which can be extracted using sequential pattern algorithms. Sequential pattern mining is a promising approach for discovering temporal regularities in huge and heterogeneous databases. These sequences represent individuals' common behavior and could contain sensitive information. Thus, sequential patterns should be sanitized to preserve individuals' privacy. Hence, many algorithms have been proposed to accomplish this task. However, these techniques add noise to the candidate support before they are validated as, frequently, and thus, they cannot be applied without having access to all the users' sequences data. In this paper, we propose a differential privacy graph-based technique for publishing frequent sequential patterns. It is applied at the post-processing stage; hence it may be used to protect frequent sequential patterns after they have been extracted, without the need to access all the users' sequences. To validate our proposal, we performed a detailed assessment of its utility as a pattern mining algorithm and calculated the impact of the sanitization mechanism on a recommender system. We further evaluated its information loss disclosure risk and performed a comparison with the *DP-FSM* algorithm.

Keywords: sequential pattern mining; differential privacy; frequent pattern mining; edge differential privacy; graph differential privacy; anonymization of big data

check for updates

Citation: Nunez-del-Prado, M.; Maehara-Aliaga, Y.; Salas, J.; Alatrista-Salas, H.; Megías, D. A Graph-Based Differentially Private Algorithm for Mining Frequent Sequential Patterns. *Appl. Sci.* **2022**, *12*, 2131. https://doi.org/10.3390/app12042131

Academic Editor: Federico Divina

Received: 24 December 2021
Accepted: 14 February 2022
Published: 18 February 2022

1. Introduction

Sequential pattern mining techniques are used to discover frequent correlations from data describing the temporal behavior of studied phenomena. Sequential pattern mining is applied in several domains, such as environmental care [1,2] or public health [3,4], among others. The datasets used for the sequential pattern mining task could contain sensitive information that could be exposed when patterns are shared. Indeed, a sequential pattern represents a frequent behavior belonging to a percentage of the individuals under study. For instance, in [5], Nuñez-del-Prado et al. studied the possibility of re-identifying an individual from a set of sequential patterns through the unicity metric. In this context, privacy-aware sequential pattern mining techniques arise to overcome this problem. If the unicity assessment of a published set of frequent sequential patterns shows that there is a risk of re-identification, such patterns may be protected through differential privacy. This contrasts with traditional methods for providing differential privacy to frequent sequential patterns since they use the entire dataset of users' sequences. The proposed method takes as input a dataset of sequences, which are processed to extract a set of frequent patterns first. Then, the method represents the relation of individuals and their behavior through a bipartite graph of users and frequent patterns. After that, the sanitization mechanism protects the bipartite graph by adding noise graphs [6] to obtain differential privacy guarantees [7]. Finally, the sequential patterns with differential privacy are obtained from the sanitized bipartite graph.

Later on, this method is compared with a baseline method called Differential Private Frequent Sequence Mining (*DP-FSM*), which introduces Laplace noise when patterns are built using the Breadth-First Search (BFS) strategy. Thus, *graph-based* and *DP-FSM*-based techniques are compared using information loss, disclosure risk, and utility metrics for the Statlog, Bank Transaction, and NYC and Tokyo Check-in datasets. We observed that our approach outperforms the baseline in terms of disclosure risk for ε values from 0.05–1.

The rest of the paper is organized as follows. Section 2 depicts the state-of-the-art about sequential pattern mining and differential privacy. Section 3 details the materials and methods used in the present effort. The main results are shown in Section 4 and discussed in Section 5. Finally, Section 6 gives the conclusion and future directions.

2. Related Works

In the present subsection, we describe the works on differential privacy for sequential pattern mining. Most of them add Laplace noise to the prefix tree while building the sequential patterns [8–11]. Chen et al. [12] applied differential privacy to publish prefix trees for transit data by adding Laplace noise to the prefix tree count at each h level. Afterward, they calculated the frequent patterns using the *PrefixSpan* algorithm on the sanitized data and measured the utility as the true and false positives for the top-k frequent sequences. They performed their experiments on data from Canada's metro and bus networks. Similarly, in [8], differential privacy was added to n-grams. They experimentally evaluated their solution in terms of the data analysis tasks of count query and frequent sequential pattern mining, with data from the page views of msnbc.com (i.e., MSNBC) and from the sequences of stations visited by passengers in the Montreal transportation system (i.e., STM). Bonomi and Xiong [13,14] applied a two-phase algorithm for differential privacy in sequential pattern mining. The first phase adds Laplace noise to release a differentially private prefix tree T'. The second phase constructs a dataset sketch and refines the candidate patterns count, finally using the remaining ε-budget to retrieve the final counts, obtaining the top-k most frequent patterns. Their utility measure was the F_1-score. Xu et al. [9,10] observed that in differentially private frequent sequence mining (FSM), the amount of required noise is proportionate to the number of candidate sequences, and they proposed the PFS^2 privacy-aware algorithm to extract frequent patterns. To improve the utility and privacy trade-off, they leveraged a sampling-based candidate pruning technique. Then, they performed the experimental evaluation on the MSNBC, House Power, and Bible datasets to compare PFS^2 with the Prefix [12] and n-gram [8] algorithms, in terms of the F-score and relative error measures. Zhou and Lin [11] proposed to truncate frequent patterns to add noise to the pattern support based on the geometric mechanism to satisfy ε-differential privacy. Differential Private Frequent Sequence Mining (DP-FSM) aims to generate all candidate sequences and add noise to each candidate sequence's support. Later on, the algorithm selects frequent sequences according to the amount of noise added to the candidate. The noise depends on the sensitivity that should be reduced for accurately estimating which candidate sequences are frequent. The sensitivity depends on the size of the sequence candidate. The authors showed that their approach outperformed the PFS^2 algorithm [10] in terms of the *false negative rate* and *relative support error*. Lee et al. [15] suggested a framework to minimize privacy leaks from medical distributed records. The first step is to extract frequent patterns from each data source. Thus, the authors applied differential privacy over the support count of the extracted patterns. Then, the frequent patterns from all local data sources were aggregated in the case and control lists. Once the lists were aggregated in a centralized location, they computed the union of patterns and carried out the summation of the support counts of each pattern. Finally, based on the aggregated list of patterns, the top-*k* patterns were chosen to be used as the feature representation.

3. Materials and Methods

In this project, we compared a new technique for adding privacy guarantees (i.e., sanitized) sequential patterns. Indeed, our proposal is based on adding noise to a bipartite graph representing, on the one hand, users and, on the other hand, their patterns. This new method was tested on three databases and compared with the Differential Private Frequent Sequence Mining (DP-FSM) [11] reference algorithm in terms of the disclosure risk and information loss measures, as well as the utility.

The following paragraphs describe the methodology performed in this effort. Fist, the sequential pattern mining is described (see Section 3.1). Later, once the patterns are obtained, they are used as the input in our proposal (see Section 3.4). Finally, our proposal is compared with a baseline algorithm [11] using three measures to validate our approach (see Section 3.5).

3.1. Sequential Pattern Mining

The sequential pattern mining problem aims to extract temporal correlations hidden from a sequences dataset. To better understand the sequential pattern mining problem, we present some key definitions in this section.

Definition 1 (Item and itemset)**.** *An* item *I is a literal value for purchase categories. An* itemset, *$IS = (I_1\ I_2 \dots I_u)$, is a non-empty set of* items *such that $I_i \in dom(items)\ \forall\ i \in [1..u-1]$.*

Definition 2 (Inclusion of itemsets)**.** *An itemset $IS = (I_1\ I_2 \dots I_u)$ is included, denoted $\subseteq$, in another itemset $IS'(I'_1\ I'_2 \dots I'_v)$, iff $\forall I_k \in IS$, $\exists\ i_k$, such that $I_k = I'_{i_k}$.*

Definition 3 (Sequence)**.** *A sequence* **S** *is an ordered list of itemsets, denoted $S = [IS_1\ IS_2 \dots IS_v]$ where IS_i, IS_{i+1} satisfy the constraint of temporal sequentiality for all $i \in [1..v-1]$.*

Definition 4 (Inclusion of sequences)**.** *A sequence $S = [IS_1 \quad IS_2 \dots IS_u]$ is included in another sequence $S' = [IS'_1 IS'_2 \dots IS'_v]$, denoted as $S \subseteq S'$, iff $\exists\ i_1 < i_2 < \dots < i_u$ such that $IS_1 \subseteq IS'_{i_1}, IS_2 \subseteq IS'_{i_2}, \dots, IS_u \subseteq IS'_{i_u}$.*

Definition 5 (Support of a sequence)**.** *We define the support of a sequence S, denoted as $supp(S)$, as the number of sequences in the database sDB that include S.*

Definition 6 (Problem of sequential pattern mining)**.** *Given a positive integer σ (minimal support) and a sequence in the database sDB, a sequence can be considered frequent if its support $supp(S)$ is greater than or equal to σ, i.e., $supp(S) \geq \sigma$. All frequent sequences are called sequential patterns, and they are stored in a pattern database pDB.*

Several algorithms have been proposed to tackle the problem of extracting sequential patterns [16–18]. These algorithms can be divided into two main strategies: Depth-First Search (DFS) and Breadth-First Search (BFS). BFS-based techniques [17] discover patterns by level, in which a pattern is built combining patterns of size k with patterns of size $k+1$. For each level, patterns should appear more than a threshold called the minimal support.

3.2. Noise Graph Addition

We consider the sequential pattern extraction here as a preprocessing step for our algorithm. After this step, we need to represent the frequent patterns and users as a bipartite graph. In [6], a formalization of graph perturbation was proposed as noise graph addition. It was shown in [7] that it provides ε-edge-differential privacy. We adapted the basic definitions and notations from [6,7] for our use-case.

Definition 7 (Bipartite graph)**.** *A bipartite graph $G = G(N, M, E)$ is defined by two sets of nodes (N and M) and a set of edges E, which are pairs of nodes ij with $i \in N$ and $j \in M$.*

Definition 8 (Graph addition)**.** *Let $G_1 = G_1(N, M, E_1)$ and $G_2 = G_2(N, M, E_2)$ be two bipartite graphs with the same sets of nodes N, M; then, the addition of G_1 and G_2 is the graph $G = G(N, M, E)$ where the edges E are defined as:*

$$E = \{e|e \in E_1 \wedge e \notin E_2\} \cup \{e|e \notin E_1 \wedge e \in E_2\}.$$

We denote G as:

$$G = G_1 \oplus G_2.$$

Therefore, after defining the graph-addition procedure, to define the bipartite noise graph mechanism, we must sample a graph from a family of random graphs and show that it is edge-differentially private.

Definition 9 (Gilbert model)**.** *The family of random bipartite graphs generated by the* Gilbert model *is the set $\mathcal{G}(n, m, p)$ of bipartite graphs $G(N, M, E)$, such that $|N| = n$, $|M| = m$ and such that for each possible pair of nodes ij, with $i \in N$ and $j \in M$, then $ij \in E$, with probability p.*

Definition 10 (Bipartite noise graph mechanism)**.** *For a bipartite graph $G = G(N, M, E)$, such that $|N| = n$, $|M| = m$. We define the bipartite noise graph mechanism $\mathcal{A}_{n,m,p}$ to be the randomization mechanism that for a given probability $\frac{1}{2} < p < 1$ outputs $\mathcal{A}_{n,m,p}(G) = E(G \oplus g)$ with $g \in \mathcal{G}(n, m, p)$.*

Theorem 1 ([7])**.** *The bipartite noise graph mechanism $\mathcal{A}_{n,m,p}$ is ε-edge-differentially private for $\varepsilon = \frac{1-p}{p}$.*

3.3. Differential Privacy

Differential privacy provides a formal guarantee that the privacy risk remains similar whether or not an individual provides his/her data [19]. It has been successfully applied to frequent sequential pattern mining, but its strict privacy guarantees oftentimes involve considerable information loss. To reduce such information loss, we propose an in-process approach to protect the relation of the patterns to users with differential privacy. Here, we include the definition for edge-differential privacy adapted from [20].

Definition 11 (Edge-differential privacy)**.** *It is said that a randomized function $\mathcal{A}$ is ϵ-*differentially private *if for all graphs G and G' differing on at most one edge and all $S \subseteq Range(\mathcal{A})$, it holds that:*

$$Pr[\mathcal{A}(G) \in S] \leq exp(\epsilon) \times Pr[\mathcal{A}(G') \in S].$$

Therefore, an adversary will not be able to infer the existence of an edge in the graph by observing the output of the function $\mathcal{A}$.

3.4. Graph-Based Differential Privacy for Frequent Sequential Patterns

Previous methods start from the sequences and provide differential privacy when extracting the frequent patterns, while the proposed method can be applied to protect frequent patterns. Hence, it does not require having access to users' sequences, only to their patterns. It consists of the following steps (as depicted in Figure 1):

1. Frequent sequential pattern mining (pre-processing);
2. Graph representation of frequent patterns;

3. Adding noise to the client-pattern graph;
4. Publishing the frequent sequential patterns with noise.

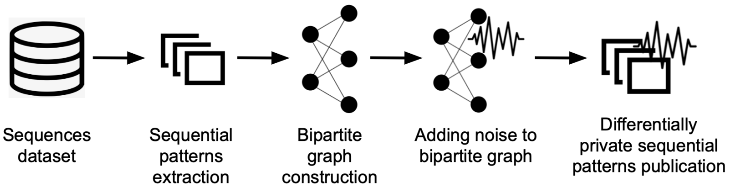

Sequences dataset
Sequential patterns extraction
Bipartite graph construction
Adding noise to bipartite graph
Differentially private sequential patterns publication

Figure 1. Process of sequential patterns' sanitization through our graph-based differentially private algorithm.

For graph representation of frequent patterns, we mine the sequential patterns from the sequences dataset to obtain the frequent **sequential patterns** pDB and associate each user i with its patterns through a dictionary $\mathcal{D} = \{client : pattern\}$. This can be considered as a pre-processing step. After extracting the frequent patterns, we generate a **bipartite graph** G from the dictionary $\mathcal{D}$, as follows: We define N as the set of clients i in the sequence database sDB and M as the set of frequent sequential patterns j in pDB and E as the pairs ij, for each client i and each pattern $j \in \mathcal{D}[i]$. Thus, $G = G(N, M, E)$ is the bipartite graph with nodes N, M and edges E. We assume that $|N| = n$ and $|M| = m$.

To add noise to the client-pattern graph, we apply the bipartite noise graph mechanism, after obtaining the graph $G = G(N, M, E)$, which provides differential privacy from Theorem 1. We obtain a **protected bipartite graph** $\tilde{G} = \mathcal{A}_{n,m,p}(G)$ as a result.

Finally, to publish the frequent sequential patterns with noise from the protected graph $\tilde{G}$, we obtain a protected client-pattern dictionary $\tilde{\mathcal{D}} = \{client : pattern\}$. From the client-pattern dictionary $\tilde{\mathcal{D}}$, we re-calculate the supports of the frequent patterns and publish the frequent sequential patterns with differential privacy guarantees; see Figure 1.

3.5. Measures

In the present subsection, we introduce different metrics to quantify information loss, disclosure risk, and utility.

Definition 12 (Information loss [8])**.** *We define the* information loss *for $\widetilde{pDB}$ as the average Relative Error (RE) of the published frequent itemset. The RE is calculated over all published frequent patterns $X \in \widetilde{pDB}$, as follows:*

$$RE(\widetilde{pDB}) = \frac{1}{|(\widetilde{pDB})|} \sum_{X \in \widetilde{pDB}} \frac{|noisy\ freq(X) - freq(X)|}{\max\{freq(X), 0.01n\}},$$

where n denotes the number of users in sDB and $0.01n$ is a sanity bound that mitigates the effect of the queries with values close to zero (as in [8]).

Definition 13 (Disclosure risk)**.** *We define the* disclosure risk *based on the Jensen–Shannon distance as:*

$$DR(\widetilde{pDB}) = 1 - JS(pDB, \widetilde{pDB}),$$

where the Jensen–Shannon distance is:

$$JS(pDB, \widetilde{pDB}) = \sqrt{\frac{D(pDB_{dist}||m) + D(\widetilde{pDB}_{dist}||m)}{2}},$$

where $\widetilde{pDB}$ is the sanitized version of the frequent sequential pattern pDB, m is the pointwise mean between pDB_{dist} and $\widetilde{pDB}_{dist}$, and D is the Kullback–Leibler divergence [21].

Concerning the utility, different metrics have been proposed in the literature. For instance, References [9,13,22] used the *F-score*, while [11] used the False Negative Rate (FNR) as defined in [12]. We consider that the *F-score* is more general than the FNR, since it combines the precision and recall, while the FNR only measures $1 - recall$.

Definition 14 (*F-score* [22])**.** *Denote as $\widetilde{pDB}$ the set of frequent itemsets generated by a differentially private frequent itemset mining algorithm, and pDB is the set of correct frequent itemsets, then:*

$$precision(\widetilde{pDB}) = \frac{|\widetilde{pDB} \cap pDB|}{|\widetilde{pDB}|},\ recall(\widetilde{pDB}) = \frac{|\widetilde{pDB} \cap pDB|}{|pDB|},$$

and:

$$F\text{-}score(\widetilde{pDB}) = \frac{2 * precision(\widetilde{pDB}) * recall(\widetilde{pDB})}{precision(\widetilde{pDB}) + recall(\widetilde{pDB})}$$

To validate the utility of the protected sequential patterns in a recommendation task, we implemented the Extended Patricia Trie (*EPT*) that takes, as input, a set of sequential patterns and a set of item weights (i.e., some value associated with an item, for instance a cost) to predict the next items. Our implementation inspired by [23] uses a Patricia trie structure to represent sequential patterns. Once the trie is built, the algorithm searches, in a depth manner, the next items as a recommendation.

To measure the utility in the recommendation task, we propose two metrics:

Definition 15 (Confidence Value (CV))**.** *To measure how accurate the recommendation is, we use the following equation:*

$$CV(SP_{test}) = \frac{1}{|SP_{test}|} \sum_{S \in SP_{test}} f(S),$$

where:

$$f(S) = \begin{cases} 1, & \textit{if the next item of } SP_{train} \in SP_{test} \\ \\ 0, & \textit{if the next item of } SP_{train} \notin SP_{test} \end{cases}$$

where SP_{train} is the sequential patterns used for training the model and PC is the set of sequences used in the testing step.

Thus, it assigns a value to the items that are correctly predicted. In this equation, if the evaluated item belongs to the set SP_{test}, then the function returns 1; otherwise, it returns 0.

Definition 16 (Coverage)**.** *This metric allows measuring if the* EPT *is sufficiently varied to make predictions of different cases. Specifically, the Coverage (CG) shows the percentage of cases that could be evaluated from the test set. The CG value is calculated as follows:*

$$CG(SP_{test}) = \frac{|SP_{train}| - UN}{|SP_{train}|},$$

where the UN variable quantifies the number of cases in the test set, where a prediction was not possible, and $|SP_{train}|$ represents the number of sequential patterns used for building the EPT.

There are two reasons for computing the coverage measure: first, when a pattern has never been seen before in the *EPT*; second, when a pattern is in the *EPT*, but does not have a suffix. It is important to measure whether the *EPT* is able to propose a pool of suitable candidates for prediction.

Regarding the impact of the sanitization mechanism in the recommendation task, we developed the following pipeline: The process begins with the sequence dataset from which the frequent patterns are extracted using the *PrefixSpan* algorithm. These frequent patterns are the input for the *graph-based* algorithm that outputs privacy-aware frequent patterns. On the other hand, the *DP-FSM* algorithm builds the privacy-aware frequent patterns directly from the sequence dataset. Thus, the patterns extracted from the *DP-FSM* and *graph-based* algorithms are used to build an *EPT* (model training step), which allows predicting the next events from the sequences. Therefore, the extracted patterns from *PrefixSpan* are used to test the *EPT* in terms of the confidence, coverage measure, and the number of cases that cannot be predicted.

4. Experiments and Results

In the present section, we describe the results of applying different measures to quantify the information loss (Section 4.1), disclosure risk (Section 4.2), and utility (Section 4.3) of the differential private sequential pattern and *DP-FSM* algorithms for three different datasets, which are described in the following paragraphs:

1. **Statlog** or **German Credit Data**: This dataset was provided by Prof. Hofmann. It contains categorical attributes to clients having good or bad credit risk (German Credit Data: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data), accessed on 13 February 2022). In this work, the characteristics of each client having credit are represented as a sequence. For instance, A40 represents if the client has a new car, and A71 represents if the client is unemployed;
2. **Bank Transactions**: this private dataset contains credit and debit card transactions in monetary units, grouped by the Classification of Individual Consumption by Purpose (COICOP), the international reference classification of household expenditure (COICOPs https://unstats.un.org/unsd/class/revisions/coicop_revision.asp, accessed on 13 February 2022). Each bank user is represented by a sequence of COICOPs (C1 to C12). For instance, C1 represents food and non-alcoholic beverages, and C3 groups clothing and footwear;
3. **NYC and Tokyo Check-in**: this dataset contains 801131 check-ins in NYC and Tokyo collected from April 2012 to February 2013. Each check-in is associated with its timestamp, GPS coordinates, and venue categories (NYC and Tokyo Check-in: https://sites.google.com/site/yangdingqi/home/foursquare-dataset#h.p_ID_46, accessed on 13 February 2022). To build the sequences, the venue categories (represented by letters) of the visited place were grouped for each user. For example, A represents a pet store, and B represents a beauty store.

The sequence <[C1, C3, C3, C5]> is an example of the bank transaction dataset. Each C_i represents one of the twelve COICOPs. In the same way, other sequences' datasets have the same shape.

Table 1 shows the main characteristics of the datasets. The number of sequences represents the number of rows in the database. The number of different items represents all the different objects in the database. The maximum and minimum sizes represent the maximum and the minimum number of items belonging to a sequence. Finally, the last column of Table 1 shows the average number of items in all sequences.

Table 1. Main characteristics of sequences' datasets.

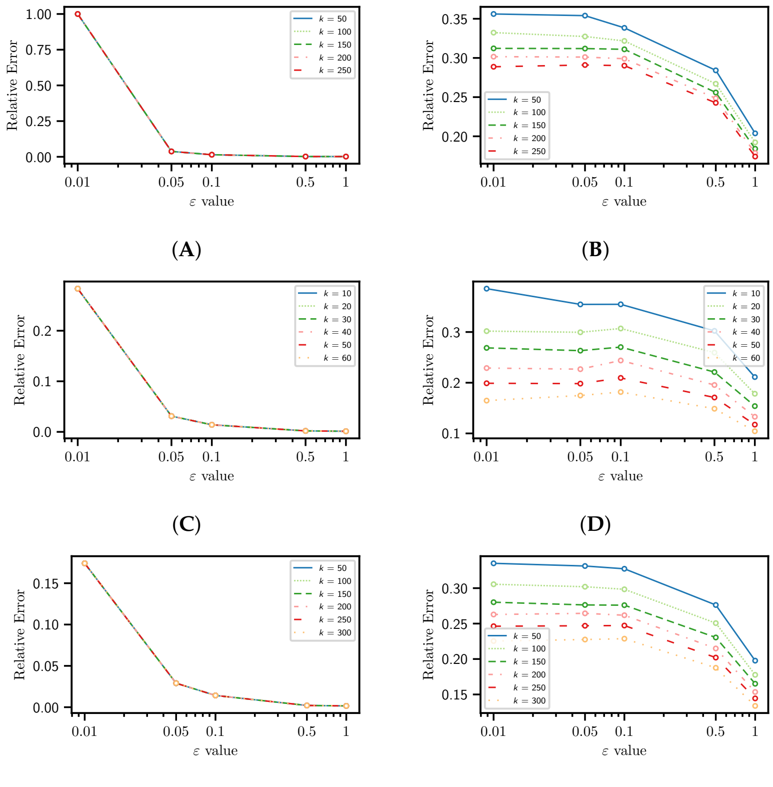
Metrics	German	Bank	NYC
Sequences	1000	548,263	1083
Different items	90	12	14
Max size	21	4028	2697
Min size	21	1	100
Avg size	21	111	210
Freq. Patterns	75	22	14,494
Rel. Support	0.5	0.8	0.9
License	Public	Private	Public

It is worth noting that, before extracting the patterns, each dataset was encoded as sequences grouping the transaction ID per timestamp. In the following subsection, we present the different metrics performed over the aforementioned datasets. We highlight that all experiments were run on a PC Intel(R) Xeon(R) 2.10 GHz with 32 cores, RAM: 200 GB. All algorithms were implemented in Ubuntu 20.04.3 LTS using Python 3 using Jupyter, Pandas, Diffprivlib, Numpy, and Networkx.

4.1. Information Loss

The two techniques compared in this paper add noise to the sequential patterns in two different manners. On the one hand, the *DP-FSM* technique adds noise to sequences as the patterns are constructed, level by level. In this technique, noise is added to the supports of the candidate sequences. If the candidate support does not reach the minimal support, it is not considered as a pattern, even though it may appear in the patterns extracted with the *PrefixSpan* algorithm. On the other hand, the *graph-based* algorithm represents the links between users and patterns. Thus, the support of a pattern is computed using the degree of the nodes representing the patterns. In this context, the privacy mechanism removes and adds links using a differential privacy mechanism causing the supports to change. Nonetheless, the privacy mechanisms cause information loss. Therefore, to quantify this loss, we rely on the relative error index, as introduced in Definition 12, for ε values 0.01, 0.1, 0.5, and 1.

Figure 2 depicts the information loss results for the different datasets. In general, we observed how the relative error decreases as ε increases. Concerning the *Bank* dataset, Figure 2A,B illustrates "for different top-k patterns" that the error using the *graph-based* algorithm decreases more smoothly. For the *German* dataset, the top extracted patterns range from 10 to 60. Besides, we note that the curves are smooth, and the top-k impacts the measure. Thus, the higher the top-k, the smaller the error is (see Figure 2D), while for the *DP-FSM*, the top-k parameter does not affect the error level, which could be a disadvantage (see Figure 2C). Finally, for the *NYC* dataset, Figure 2E,F shows the highest number of extracted patterns from 50 to 300. In that case, we observed a different behavior for the *DP-FSM* algorithm influenced by the top-k parameter. Therefore, even if the *DP-FSM* algorithm introduces less error, it needs hundreds of extracted patterns to present different performances affected by the top-k parameter.



(**A**) (**B**)

(**C**) (**D**)

(**E**) (**F**)

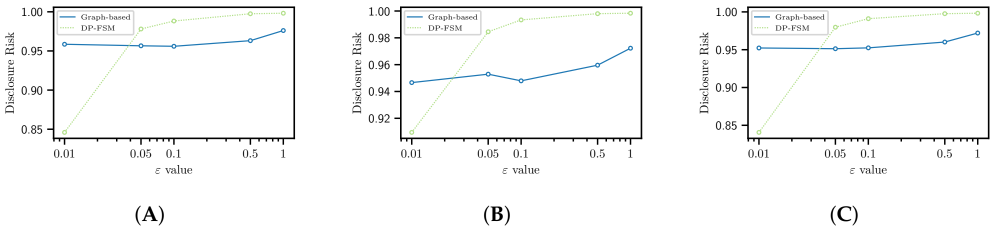
Figure 2. Information loss measured as the relative error. Where (**A**) *DP-FSM* Bank, (**B**) *Graph-based* Bank, (**C**) *DP-FSM* German, (**D**) *Graph-based* German, (**E**) *DP-FSM* NYC, and (**F**) *Graph-based* NYC.

4.2. Disclosure Risk

To measure the disclosure risk, we compare the support distribution difference between the patterns obtained without a privacy mechanism, using the classical *PrefixSpan* algorithm, compared to the patterns obtained by the privacy mechanisms described above. Therefore, this metric quantifies the certainty of the obtained patterns.

Thus, if the patterns are known with high certainty, it can be assumed that the adversary's knowledge is more accurate to perform an inference attack. To assess the disclosure risk, we used the Jensen–Shannon divergence, which is computed between the support distributions of the patterns obtained by the *PrefixSpan*, *DP-FSM*, and *graph-based* algorithms. Intuitively, when the Jensen–Shannon divergence is close to zero, the difference between the patterns with and without noise is minimal, and the disclosure risk is maximized. Conversely, the greater the Jensen–Shannon divergence, the greater the difference between the patterns with and without noise is, minimizing the disclosure risk.

Figure 3 depicts the results of the disclosure risk assessment. In this case, we considered all the extracted patterns to calculate the Jensen–Shannon divergence. Figure 3A shows that the DR is larger for the *DP-FSM* than for the *graph-based* algorithm, for ε values from $\varepsilon = 0.05$, while it is smaller only for $\varepsilon = 0.01$ for the German and NYC datasets. This illustrates that our mechanism improves the modulation of the disclosure risk.



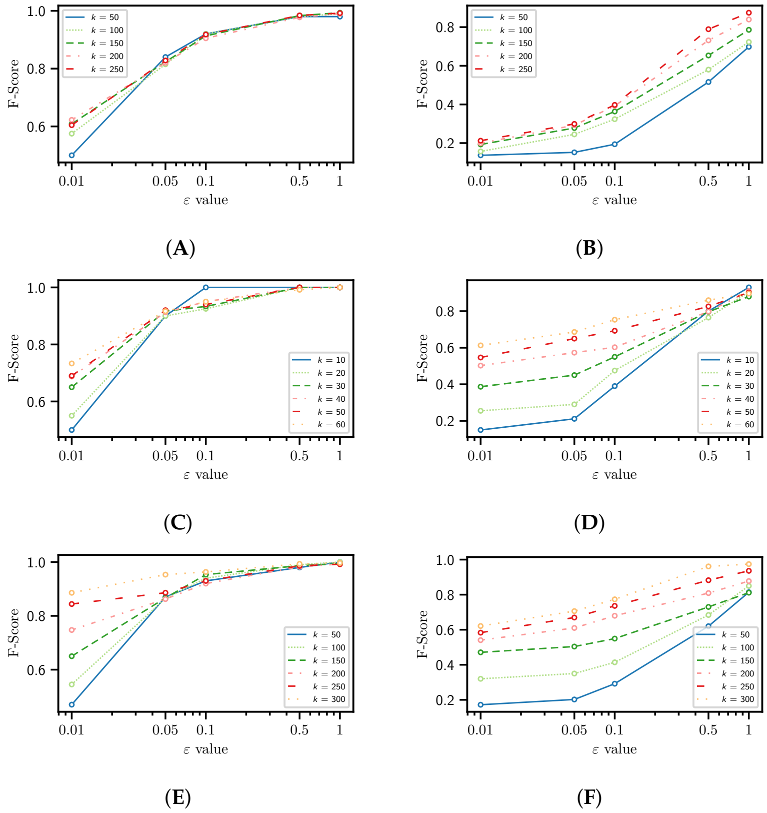
(**A**) (**B**) (**C**)

Figure 3. Disclosure risk measures' comparison between the *graph-based* and *DP-FSM* algorithms. Where (**A**) is the Bank dataset, (**B**) is the German dataset, and (**C**) NYC dataset.

4.3. Utility

In this section, we measure the performance of a differentially private frequent pattern mining algorithm on finding the top-*k* most frequent patterns. For this, we considered the F-score classification measure.

Figure 4 illustrates the F-score obtained from the experiments using the *DP-FSM* and *graph-based* algorithms. We note in Figure 4A that the *F-score* is not affected by the ε parameter for the *DP-FSM* algorithm in the *Bank* dataset, while in Figure 4B, we observe the influence of the parameter over the *F-score*. The same behavior is depicted for the *Bank* and *German* datasets. Besides, we observe the influence of the top-*k* parameter over the *F-score*: the smaller the *k* value, the lower the *F-score* is.



(**A**) (**B**)

(**C**) (**D**)

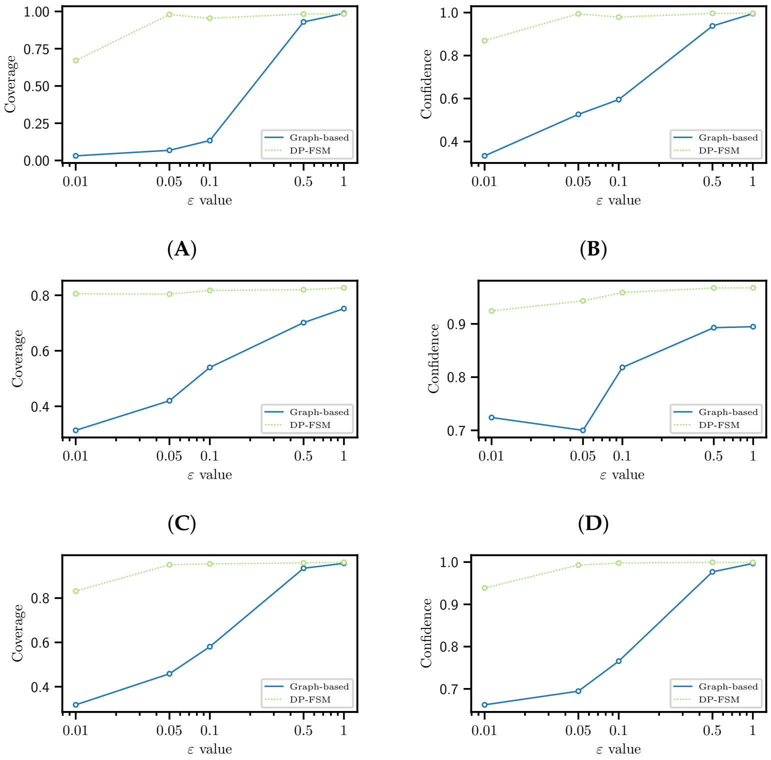
(**E**) (**F**)

Figure 4. Utility measures using the F-score for *DP-FSM* and *Graph-based* applied to different dataset. Where (**A**) *DP-FSM* Bank, (**B**) *Graph-based* Bank, (**C**) *DP-FSM* German, (**D**) *Graph-based* German, (**E**) *DP-FSM* NYC, and (**F**) *Graph-based* NYC.

4.4. Utility for Recommendation Tasks

To complement the *F-score* metric, we quantified the utility of the *graph-based* method through a recommendation task algorithm. This section describes the process, the technique, and the main results.

We performed a recommendation using the *EPT* algorithm from two methods of protecting a sequential pattern dataset as the input: (1) graph-based differentially private publication of sequential patterns and (2) differentially private sequential pattern mining through the *DP-FSM* algorithm. Both recommendations sets were compared with the results using classical patterns (without noise) to measure the utility for the recommendation of both sanitization techniques. Two metrics were used for this task: Confidence and Coverage values. The first metric allows us to perceive how reliable the prediction is using the constructed EPT, besides the second measure will enable us to realize if the EPT is sufficiently varied to make predictions of different cases. Figure 5 shows the coverage and confidence measures for the recommendations' comparison between the graph-based and the DP-FSM algorithms for the three datasets at our disposal. They show that for ε values smaller than one, the *DP-FSM* algorithm preserves better the metrics than the graph-based algorithm, which equalizes its performance from values $\varepsilon = 1$.



(**A**) (**B**)

(**C**) (**D**)

(**E**) (**F**)

Figure 5. Coverage and confidence measures for the recommendations' comparison between the *graph-based* and *DP-FSM* algorithms. Where (**A**) Coverage Bank, (**B**) Confidence Bank, (**C**) Coverage German, (**D**) Confidence German, (**E**) Coverage NYC, and (**F**) Confidence NYC.

5. Discussion

A sequence represents the temporal correlation of a set of features describing a phenomenon (such as a customer's purchases at different times). From a set of sequences, the task of sequence pattern mining allows extracting the sub-sequences that appear frequently called sequence patterns. In this paper, a new *graph-based* technique to add noise to sequential patterns is proposed. This technique builds a bipartite graph of users and patterns. Later on, a ε-differential privacy mechanism removes or adds edges between users and patterns, making the patterns change their support (i.e., pattern nodes' degree). This new technique was compared with the *DP-FSM* algorithm, which adds noise to the candidate sequences before they become frequent. Since both techniques add noise to a sequence in different ways, the aim of this work was to compare them on three different datasets through different measures. In addition, both algorithms were compared in terms of utility using a sequence-based recommendation algorithm. The main results showed that our proposal outperformed the baseline algorithm *DP-FSM*.

An advantage of the *graph-based* technique is the noise addition at the post-processing stage instead of at the extraction stage, as the *DP-FSM* method. In detail, with the *graph-based* technique, it is possible to sanitize frequent pattern data, while with *DP-FSM*, it is not: *DP-FSM* needs to access all the sequences data and possibly more sensitive data. Besides, the *graph-based* technique uses coarse-grained data (i.e., frequent patterns), which could even be produced by a third party. Therefore, this third party could have access to the sequences to produce the input patterns to be sanitized in an agnostic way, through the bipartite graph, without knowing what the sequences in the patterns represent.

The limitation of our approach is the dependency of an algorithm to extract patterns. Thus, the frequent pattern extraction performance will impact the accomplishment of the sanitization process.

One of the important characteristics of our approach is that it enables performing users/patterns and patterns' sanitizations. The former sanitization lets the data curator report the patterns by users providing a privacy guarantee depending on the noise addition level of the ε-differential privacy mechanism. The latter allows reporting only the sanitized support of patterns, which provides stronger privacy by adding an aggregation step after the output of the ε-differential privacy mechanism.

6. Conclusions

In this work, we proposed a *graph-based* sanitization algorithm for frequent sequential patterns. We compared our approach to the baseline sanitization algorithm for extracting privacy-aware frequent patterns (*DP-FSM*) considering the information loss, disclosure risk, and utility measures. Besides, we quantified the impact of the privacy mechanisms for a recommendation task. We observed that our approach outperformed the baseline in terms of disclosure risk for ε values from 0.05–1. Furthermore, we showed that it provides more flexibility for reporting users/patterns of patterns. As new research avenues, we will analyze the constructed trees in the *EPT* to assess whether an adversary could extract some knowledge about the dataset used to build the tree.

Author Contributions: Conceptualization, M.N.-d.-P., J.S. and H.A.-S.; methodology, M.N.-d.-P., J.S. and H.A.-S.; software, Y.M.-A.; validation, M.N.-d.-P., J.S., Y.M.-A. and H.A.-S.; formal analysis, J.S.; investigation, M.N.-d.-P., J.S. and H.A.-S.; resources, D.M.; data curation, Y.M.-A.; writing—original draft preparation, all authors; writing—review and editing, all authors; visualization, M.N.-d.-P., J.S. and Y.M.-A.; supervision, M.N.-d.-P., J.S. and H.A.-S.; project administration, D.M.; funding acquisition, D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Spanish Government under Projects RTI2018-095094-B-C21 and RTI2018-095094-B-C22 "CONSENT".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alatrista-Salas, H.; Azé, J.; Bringay, S.; Cernesson, F.; Selmaoui-Folcher, N.; Teisseire, M. A knowledge discovery process for spatiotemporal data: Application to river water quality monitoring. *Ecol. Inform.* **2015**, *26*, 127–139. [CrossRef]
2. Zhang, L.; Yang, G.; Li, X. Mining sequential patterns of PM2.5 pollution between 338 cities in China. *J. Environ. Manag.* **2020**, *262*, 110341. [CrossRef] [PubMed]
3. Pinaire, J.; Chabert, E.; Azé, J.; Bringay, S.; Poncelet, P.; Landais, P. Prediction of In-Hospital Mortality from Administrative Data: A Sequential Pattern Mining Approach. *Stud. Health Technol. Inform.* **2021**, *281*, 293–297. [PubMed]
4. Tandan, M.; Acharya, Y.; Pokharel, S.; Timilsina, M. Discovering symptom patterns of COVID-19 patients using association rule mining. *Comput. Biol. Med.* **2021**, *131*, 104249. [CrossRef] [PubMed]

5. Nunez-del Prado, M.; Salas, J.; Alatrista-Salas, H.; Maehara-Aliaga, Y.; Megías, D. Are Sequential Patterns Shareable? Ensuring Individuals' Privacy. In *International Conference on Modeling Decisions for Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 28–39.
6. Torra, V.; Salas, J. Graph Perturbation as Noise Graph Addition: A New Perspective for Graph Anonymization. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*; Springer International Publishing: Cham, Switzerland, 2019; pp. 121–137.
7. Salas, J.; Torra, V. Differentially Private Graph Publishing and Randomized Response for Collaborative Filtering. In Proceedings of the 17th International Joint Conference on e-Business and Telecommunications, ICETE 2020-V2: SECRYPT, Lieusaint, Paris, France, 8–10 July 2020; ScitePress: Setúbal, Portugal, 2020; pp. 415–422.
8. Chen, R.; Acs, G.; Castelluccia, C. Differentially private sequential data publication via variable-length n-grams. In Proceedings of the 2012 ACM Conference on Computer and Communications Security, Raleigh, NC, USA, 16–18 October 2012; pp. 638–649.
9. Xu, S.; Cheng, X.; Su, S.; Xiao, K.; Xiong, L. Differentially private frequent sequence mining. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 2910–2926. [CrossRef]
10. Xu, S.; Su, S.; Cheng, X.; Li, Z.; Xiong, L. Differentially private frequent sequence mining via sampling-based candidate pruning. In Proceedings of the 2015 IEEE 31st International Conference on Data Engineering, Seoul, Korea, 13–17 April 2015; pp. 1035–1046.
11. Zhou, F.; Lin, X. Frequent sequence pattern mining with differential privacy. In *International Conference on Intelligent Computing*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 454–466.
12. Chen, R.; Fung, B.C.; Desai, B.C.; Sossou, N.M. Differentially private transit data publication: A case study on the montreal transportation system. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 12–16 August 2012; pp. 213–221.
13. Bonomi, L.; Xiong, L. A two-phase algorithm for mining sequential patterns with differential privacy. In Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, San Francisco, CA, USA, 27 October–1 November 2013; pp. 269–278.
14. Bonomi, L.; Xiong, L. Mining frequent patterns with differential privacy. *Proc. VLDB Endow.* **2013**, *6*, 1422–1427. [CrossRef]
15. Lee, E.W.; Xiong, L.; Hertzberg, V.S.; Simpson, R.L.; Ho, J.C. Privacy-preserving Sequential Pattern Mining in distributed EHRs for Predicting Cardiovascular Disease. *AMIA Jt. Summits Transl. Sci. Proc.* **2021**, *2021*, 384–393. [PubMed]
16. Pei, J.; Han, J.; Mortazavi-Asl, B.; Wang, J.; Pinto, H.; Chen, Q.; Dayal, U.; Hsu, M.C. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1424–1440.
17. Agrawal, R.; Srikant, R. Fast algorithms for mining association rules. In Proceedings of the 20th VLDB Conference, Santiago, Chile, 12 September 1994; Volume 1215, pp. 487–499.
18. Alatrista-Salas, H.; Guevara-Cogorno, A.; Maehara, Y.; Nunez-del Prado, M. Efficiently Mining Gapped and Window Constraint Frequent Sequential Patterns. In *International Conference on Modeling Decisions for Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 240–251.
19. Dwork, C. Differential Privacy. In Proceedings of the 33rd International Conference on Automata, Languages and Programming-Volume Part II (ICALP'06), Venice, Italy, 10–14 July 2006; pp. 1–12.
20. Hay, M.; Li, C.; Miklau, G.; Jensen, D. Accurate Estimation of the Degree Distribution of Private Networks. In Proceedings of the 2009 Ninth IEEE International Conference on Data Mining, Miami Beach, FL, USA, 6–9 December 2009; pp. 169–178.
21. Van Erven, T.; Harremos, P. Rényi divergence and Kullback–Leibler divergence. *IEEE Trans. Inf. Theory* **2014**, *60*, 3797–3820. [CrossRef]
22. Zeng, C.; Naughton, J.F.; Cai, J.Y. On differentially private frequent itemset mining. *Proc. VLDB Endow.* **2012**, *6*, 25–36. [CrossRef]
23. Suneetha, K.; Rani, M.U. Web Page Recommendation Approach Using Weighted Sequential Patterns and Markov Model. *Glob. J. Comput. Sci. Technol.* **2012**, *1–12* . Available online: https://computerresearch.org/index.php/computer/article/view/493 (accessed on 13 February 2022).