Mining Roles From Event Logs While Preserving Privacy

Majid Rafiei[0000−0001−7161−6927] and Wil M.P. van der Aalst[0000−0002−0955−6940]

Chair of Process and Data Science, RWTH Aachen University, Aachen, Germany

Abstract. Process mining aims to provide insights into the actual processes based on event data. These data are widely available and often contain private information about individuals. On the one hand, knowing which individuals (known as resources) performed specific activities can be used for resource behavior analyses like *role mining* and is indispensable for bottleneck analysis. On the other hand, event data with resource information are highly *sensitive*. Process mining should reveal insights in the form of annotated models, but should not reveal sensitive information about individuals. In this paper, we show that the problem cannot be solved by naïve approaches like encrypting data, and an anonymized person can still be identified based on a few well-chosen events. We, therefore, introduce a *decomposition* method and a collection of techniques that preserve the privacy of the individuals, yet, at the same time, roles can be discovered and used for further bottleneck analyses without revealing sensitive information about individuals. To evaluate our approach, we have implemented an interactive environment and applied our approach to several real-life and artificial event logs.

Keywords: Responsible process mining · Privacy preserving · Social network discovery · Role mining · Process mining

1 Introduction

In recent years, process mining has emerged as a field which bridges the gap between data science and process science [1]. Event logs are used by process mining algorithms to extract and analyze the real processes. An event log is a collection of events and such information is widely available in current information systems [3]. Each event is described by its attributes and some of them may refer to individuals, i.e., human actors. The *resource* attribute may refer to the person performing the corresponding activities [1]. Organizational process mining is a sub-discipline of process mining focusing on resource behavior using the resource attributes of events. This form of process mining can be used to extract the roles in a process or organization [4]. A simple example is when two resources perform the same set of activities, the same role can be assigned to them. Moreover, resource information is essential for bottleneck analysis and for finding the root causes of performance degradation.

Event data contain highly sensitive information and when the individuals' data are included, privacy issues become more challenging. As discussed in [9], event data may lead to privacy breaches. In addition, data protection regulations like the European General Data Protection Regulation (GDPR) impose many challenges and concerns regarding processing of personal data. In this paper, we show that preserving privacy in process mining cannot be provided by naïve approaches like encryption/anonymization and presence of some implicit information together with background knowledge can be exploited to deduce sensitive data even from minimized encrypted data.

We present a privacy-aware approach to discover roles from event logs. A *decomposition* method along with some techniques are introduced to protect the private information of the individuals in event data against frequency-based attacks in this specific context. The discovered roles can be replaced by the resources and utilized for bottleneck analyses while personal identifiers do not need to be processed anymore. We evaluate our approach w.r.t the typical trade-off between privacy guarantees and loss of accuracy. To this end, the approach is evaluated on multiple real-life and synthetic event logs.

The rest of the paper is organized as follows. Section 2 outlines related work. In Section 3, the main concepts are briefly described. In Section 4, the problem is explored in detail. We explain our approach in Section 5. In Section 6, the implementation and evaluation are described, and Section 7 concludes the paper.

2 Related Work

During the last decade, confidentiality and privacy-preserving challenges have received increasing attention. In data science, many privacy algorithms have been presented which cover topics ranging from *privacy quantification* to *downgrading the results* [5]. These algorithms aim to provide privacy guarantees by different methods, e.g., k-anonymity, l-diversity, and t-closeness [8] are series of algorithms having been presented with the initial idea that *each individual should not be distinguished from at least* $k-1$ *other individuals*.

Recently, there have been lots of breakthroughs in process mining ranging from *process discovery* and *conformance checking* to *performance analysis*. However, the research field confidentiality and privacy has received rather little attention, although the *Process Mining Manifesto* [3] also points out the importance of privacy. *Responsible Process Mining* (RPM) [2] is the sub-discipline focusing on possible negative side-effects of applying process mining. RPM addresses concerns related to Fairness, Accuracy, Confidentiality, and Transparency (FACT). In [9], the aim is to provide an overview of privacy challenges in process mining in human-centered industrial environments. A method for securing event logs to conduct process discovery by Alpha algorithm has been proposed by [11]. In [6], a possible approach toward a solution, allowing the outsourcing of process mining while ensuring the confidentiality of dataset and processes, has been presented. In [7], the aim is to apply k-anonymity and t-closeness on event data while the assumed background knowledge is a prefix of the sequence of activi-

ties. In [10], a framework has been introduced, which provides a generic scheme for confidentiality in process mining. In this paper, for the first time, we focus on the organizational perspective of event data.

3 Preliminaries: Process Mining and Role Mining

In this section, we define basic concepts regarding process mining and discovering social networks from event logs which in turn are used for role mining.

3.1 Process Mining

An event log is a collection of traces, each represented by a sequence of events. For a given set A. A^* is the set of all finite sequences over A, and $\mathcal{B}(A^*)$ is the set of all multisets over the set A^*. A finite sequence over A of length n is a mapping $\sigma \in \{1, ..., n\} \rightarrow A$, represented by a string, i.e., $\sigma = \langle a_1, a_2, ..., a_n \rangle$ where $\sigma_i = a_i$ for any $1 \leq i \leq n$. $|\sigma|$ denotes the length of the sequence. Also, $set(\sigma) = \{a \mid a \in \sigma\}$, e.g., $set(\langle a, b, c, c, b \rangle) = \{a, b, c\}$, and $multiset(\sigma) = [a \mid a \in \sigma]$, e.g., $multiset(\langle a, b, c, c, b \rangle) = [a, b^2, c^2]$.

Definition 1 (Event). *An event is a tuple $e = (a, r, c, t, d_1, ..., d_m)$, where $a \in \mathcal{A}$ is the activity associated with the event, $r \in \mathcal{R}$ is the resource, who is performing the activity, $c \in \mathcal{C}$ is the case id, $t \in \mathcal{T}$ is the event timestamp, and d_1,...,d_m is a list of additional attributes values, where for any $1 \leq i \leq m, d_i \in \mathcal{D}_i$ (domain of attributes). We call $\xi = \mathcal{A} \times \mathcal{R} \times \mathcal{C} \times \mathcal{T} \times \mathcal{D}_1 \times ... \times \mathcal{D}_m$ the event universe. An event log is a subset of ξ where each event can appear only once, and events are uniquely identifiable by their attributes.*

Definition 2 (Simple Event Log). *A simple event log $EL \in \mathcal{B}((\mathcal{R} \times \mathcal{A})^*)$ is a multiset of traces. A trace $\sigma \in EL$ is a sequence of events $\sigma = \langle (r_1, a_1), (r_2, a_2), ..., (r_n, a_n) \rangle$ where each event is represented by a resource r_i and activity a_i. Also, $set(EL) = \{set(\sigma) \mid \sigma \in EL\}$, and $multiset(EL) = [multiset(\sigma) \mid \sigma \in EL]$.*

Definition 3 (Activities and Resources of Event Log). *Let $EL \in \mathcal{B}((\mathcal{R} \times \mathcal{A})^*)$ be an event log, $act(EL) = \{a \in \mathcal{A} \mid \exists_{\sigma \in EL} \exists_{r \in \mathcal{R}} (r, a) \in \sigma\}$ is the set of activities in the event log, and $res(EL) = \{r \in \mathcal{R} \mid \exists_{\sigma \in EL} \exists_{a \in \mathcal{A}} (r, a) \in \sigma\}$ is the set of resources in the event log.*

Table 1 shows an event log, where *Case ID*, *Timestamp*, *Activity*, *Resource*, and *Cost* are the attributes. Each row represents an event, e.g., the first row shows that activity "Register" was done by resource "Frank" at time "01-01-2018:08.00" for case "1" with cost "1000". In the remainder, we will refer to the *activities* and the *resources* of Table 1 with their abbreviations.

Definition 4 (Frequencies). *Let $EL \in \mathcal{B}((\mathcal{R} \times \mathcal{A})^*)$ be an event log. The frequency of an activity a is $\#_a(EL) = \sum_{\sigma \in EL} |[(r, a') \in \sigma \mid a' = a]|$, the set of the activity frequencies is $frq(EL) = \{(a, \#_a(EL)) \mid a \in act(EL)\}$. $\#_{most}(EL)$ is the highest frequency, $\#_{least}(EL)$ is the lowest frequency, $\#_{median}(EL)$ is the median of frequencies, and $\#_{sum}(EL)$ is the sum of frequencies.*

Table 1: Sample event log (each row represents an event).

Case ID	Timestamp	Activity	Resource	Cost
1	01-01-2018:08.00	Register (R)	Frank (F)	1000
2	01-01-2018:10.00	Register (R)	Frank (F)	1000
3	01-01-2018:12.10	Register (R)	Joey (J)	1000
3	01-01-2018:13.00	Verify-Documents (V)	Monica (M)	50
1	01-01-2018:13.55	Verify-Documents (V)	Paolo (P)	50
1	01-01-2018:14.57	Check-Vacancies (C)	Frank (F)	100
2	01-01-2018:15.20	Check-Vacancies (C)	Paolo (P)	100
4	01-01-2018:15.22	Register (R)	Joey (J)	1000
2	01-01-2018:16.00	Verify-Documents (V)	Frank (F)	50
2	01-01-2018:16.10	Decision (D)	Alex (A)	500
5	01-01-2018:16.30	Register (R)	Joey (J)	1000
4	01-01-2018:16.55	Check-Vacancies (C)	Monica (M)	100
1	01-01-2018:17.57	Decision (D)	Alex (A)	500
3	01-01-2018:18.20	Check-Vacancies (C)	Joey (J)	50
3	01-01-2018:19.00	Decision (D)	Alex (A)	500
4	01-01-2018:19.20	Verify-Documents (V)	Joey (J)	50
5	01-01-2018:20.00	Special-Case (S)	Katy (K)	800
5	01-01-2018:20.10	Decision (D)	Katy (K)	500
4	01-01-2018:20.55	Decision (D)	Alex (A)	500

In the following, we define the sensitive frequencies on the basis of the box plot of the frequencies in such a way that not only the outliers but also all the other unusual frequencies are classified as sensitive. The activities having the sensitive frequencies are more likely to be identified by an adversary.

Definition 5 (Bounds of Frequencies). *Let $EL \in \mathcal{B}((\mathcal{R} \times \mathcal{A})^*)$ be an event log. We define $upper(EL) = \langle \#_a(EL) \mid \#_a(EL) > upper\_quartile \rangle$ and $lower(EL) = \langle \#_a(EL) \mid \#_a(EL) < lower\_quarile \rangle$ as the bounds of frequencies on the basis of the box plot of the frequencies such that for any $1 \leq i \leq |upper(EL)|-1$, $upper_i(EL) \geq upper_{i+1}(EL)$, and for any $1 \leq i \leq |lower(EL)|-1$, $lower_i(EL) \leq lower_{i+1}(EL)$.*

Definition 6 (Gaps). *Let $EL \in \mathcal{B}((\mathcal{R} \times \mathcal{A})^*)$ be an event log. For each bound of the frequencies, $gap_{bound}(EL) = [|bound_i(EL) - bound_{i+1}(EL)| \mid 1 \leq i \leq |bound(EL)|-1]$, and $mean(gap_{bound}(EL))$ is the mean of the gaps.*

Definition 7 (Sensitive Frequencies). *Let $EL \in \mathcal{B}((\mathcal{R} \times \mathcal{A})^*)$ be an event log. For each bound of the frequencies, $sstv_{bound}(EL) = [bound_i(EL) \mid \forall_{1 \leq i \leq |bound(EL)|-1} |bound_i(EL) - bound_{i+1}(EL)| \leq mean(gap_{bound}(EL))]$. If $|sstv_{bound}(EL)| = |bound(EL)|-1$, $sstv_{bound}(EL) = \emptyset$, i.e., there is no gap greater than the mean of the gaps. Also, $act(sstv_{bound}(EL)) = \{a \in act(EL) \mid \#_a(EL) \in sstv_{bound}(EL)\}$.*

3.2 Role Mining

When discovering a process model from an event log, the focus is on the process activities and their dependencies. When deriving roles and other organizational entities, the focus is on the relation between individuals based on their activities. The metrics based on *joint activities*, used for discovering roles and organization structures, consider each individual as a vector of activity frequencies performed by the individual and use a similarity measure to calculate the similarity between two vectors. A social network is constructed between individuals such

that if the similarity is greater than a minimum threshold (Θ), the corresponding individuals are connected with an undirected edge. The individuals in the same connected part are supposed to play the same role [4].

Consider Table 1 and let us assume that the order of the activities in each vector is D, V, C, R, S. Then, Paolo's vector is $P = (0, 1, 1, 0, 0)$, and Monica's vector is $M = (0, 1, 1, 0, 0)$. Therefore, the similarity between these vectors is 1. In this paper, we use a *Resource-Activity Matrix* (RAM), which is defined as follows, as a basis for extracting the vectors and deriving roles.

Definition 8 (Resource-Activity Matrix (RAM)). *Let $EL \in \mathcal{B}((\mathcal{R} \times \mathcal{A})^*)$ be an event log, $a \in act(EL)$, and $r \in res(EL)$: $RAM_{EL}(r, a) = \sum_{\sigma \in EL} |[x \in \sigma \mid x = (r, a)]|$, and $RAM_{EL}(r) = (RAM_{EL}(r, a_1), RAM_{EL}(r, a_2), ..., RAM_{EL}(r, a_n))$, where n is the number of unique activities.*

Table 2 shows the RAM derived from Table 1. Given the RAM, the *joint-activities* social network can be obtained as follows.

Definition 9 (Joint-Activities Social Network (JSN)). *Let $EL \in \mathcal{B}((\mathcal{R} \times \mathcal{A})^*)$ be an event log, RAM_{EL} be a resource-activity matrix resulting from the EL, and $sim(r_1, r_2)$ be a similarity relation based on the vectors $RAM_{EL}(r_1)$ and $RAM_{EL}(r_2)$, $JSN_{EL} = (res(EL), E)$ is the joint-activities social network, where $E = \{(r_1, r_2) \in res(EL) \times res(EL) \mid sim(r_1, r_2) > \Theta\}$ is the set of undirected edges between resources, and Θ is the threshold of similarities.*

Note that various similarity measures are applicable, e.g., Euclidean, Jaccard, Pearson, etc. Figure 1 shows the network and roles having been obtained by applying threshold 0.1 when using *Pearson* as the similarity measure.

4 The Problem (Attack Analysis)

Here, we discuss the general problem of confidentiality/privacy in process mining, then we focus on the specific problem and the attack model w.r.t. this research.

Table 2: The RAM from Table 1

	D	V	C	R	S
Frank	0	1	1	2	0
Joey	0	1	1	3	0
Alex	4	0	0	0	0
Katy	1	0	0	0	1
Paolo	0	1	1	0	0
Monica	0	1	1	0	0

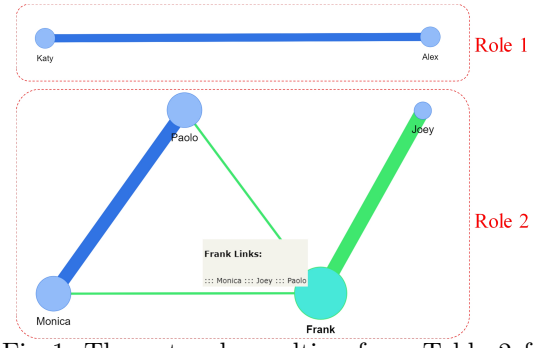


Fig. 1: The network resulting from Table 2 for Pearson similarity 0.1

4.1 General Problem

Consider Table 3 as an entirely encrypted event log with information about surgeries. The standard attributes (*Case ID*, *Activity*, *Resource*, and *Timestamp*) are included. Process mining techniques need to preserve differences. Hence, *Case ID*, *Activity*, and *Resource* are encrypted based on a deterministic encryption method.[1] Numerical data (i.e., *Timestamp*) are encrypted using a homomorphic encryption method so that the basic mathematical computations can be applied. Although the fully encrypted event log seems secure, it is not.

One can find the most or the least frequent activities and given background knowledge, the encrypted values can be simply replaced with the real values. In addition, the position of activities can also be used to infer sensitive information, e.g., when an activity is always the first/last activity, given domain knowledge the real activity can be deduced. These kinds of attacks are considered as *frequency-based*. Note that the corresponding performers are most likely identifiable, after inferring the actual activity names.

In addition to the above-mentioned attacks, other attributes are also exploitable to identify the actual activities and resources. For example, when *timestamp* is encrypted by a deterministic homomorphic encryption method, then the duration between two events is derivable. Based on background knowledge, one can infer that the longest/shortest duration belongs to specific events. When there are more attributes, it is more likely that one can combine these to infer other attributes.

Table 3: An encrypted event log.

Case ID	Activity	Resource	Timestamp
rt!@45	kl56^*	lo09(kl	3125
rt!@45	bn,.^q	lo09(kl	3256
)@!1yt	kl56^*	lo09(kl	4879
)@!1yt	bvS(op	/.,ldf	5214
)@!1yt	jhg!676	nb][,b]	6231
er^7*	kl56^*	lo09(kl	6534
er^7*	2ws34S	v,[]df	7230

These examples clarify that given domain knowledge, data leakage is possible even from a basic event log which is totally encrypted. Moreover, if the mining techniques are applied to encrypted event logs, the results are also encrypted, and data analyst is not able to interpret them without decryption [10].

4.2 Attack Analysis

Now, let us focus on our specific context where the aim is to extract roles without revealing *who performed what?* As described in Section 3, roles can be derived from a simple event log, and the *activity* is considered as the *sensitive attribute* in this setting. Therefore, activities get hashed, and we define $H(\mathcal{A})$ as universe of hashed activities ($H(X) = \{H(x) \mid x \in X\}$).[2]

We assume the frequencies of activities as background knowledge (bk) which can be formalized as $bk \in \mathcal{P}_{NE}(\mathbb{U}_{frq}) \times \mathcal{P}_{NE}(H(\mathcal{A})) \rightarrow \mathcal{P}(\mathcal{A})$, where $\mathbb{U}_{frq} = H(\mathcal{A}) \times \mathbb{N}$ is the universe of the hashed activity frequencies, and $\mathcal{P}_{NE}(X)$ is the set of all non-empty sets over the set X. Therefore, the actual activities

[1] A deterministic cryptosystem produces the same ciphertext for a given plaintext and key.

[2] H is a one-way hash function, here we use *SHA-256*.

can be revealed based on the assumed background knowledge. For example, in the event log Table 1, the least frequent activity is "Special-Case" which can be revealed based on background knowledge regarding the frequencies. We consider this information disclosure as *activity disclosure* (kind of *attribute disclosure*). Note that resources are usually not the unique identifiers in event logs. Nevertheless, they could get encrypted or hashed. Here, our focus is on activities, and the challenge is to eliminate the frequency of activities, while they are necessary to measure the similarity of resources and deriving roles. Our approach also improves privacy when background knowledge is about traces, e.g., length of traces and the position of activities in traces.

5 Approach

The idea is to decompose activities into other activities such that the *frequency* and *position* of activities get perturbed. However, at the same time, the similarities between resources should remain as similar as possible. To this end, we need to determine the number of substitutions for each activity, and the way of distributing the frequency of the main activity among its substitutions. We consider $D(H(\mathcal{A}))$ as the universe of hashed activities after the decomposition, and the sanitized event logs are obtained as follows.

Definition 10 (Sanitized Event Logs (EL''_t, EL''_{ms}, and EL''_s)). *Let $EL' \in \mathcal{B}((\mathcal{R} \times H(\mathcal{A}))^*)$ be an event log where activity names are hashed, and $Decom \in H(\mathcal{A}) \to D(H(\mathcal{A}))$ be a decomposition method. $EL''_t \in \mathcal{B}((\mathcal{R} \times D(H(\mathcal{A})))^*)$ is a trace-based sanitized event log. A multiset-based sanitized event log is $EL''_{ms} = multiset(EL''_t)$, and a set-based sanitized event log is $EL''_s = set(EL')$.*

EL''_s is used when the similarity measure is binary (Jaccard, hamming, etc.). In this case, the frequencies could be simply ignored, since these measures do not consider the absolute frequency but only whether it is 0 or not. EL''_{ms} is employed when traces are not needed to be reconstructed from the sanitized event log. In this case, the sanitized event log entirely preserves privacy of the individuals against attribute disclosure when background knowledge is trace-based. Moreover, it is clear that resource-activity matrices and the corresponding joint-activities social networks can be simply derived from the sanitized event logs. In the remainder, we use EL' for the event log where activity names are hashed and EL'' for the sanitized event logs made by applying the decomposition method, i.e., EL''_t and EL''_{ms}.

5.1 Decomposition Method

The Number of Substitutions for each activity a (NS_a) should be specified in such a way that the activities having the sensitive frequencies are not certainly identifiable anymore. In the following, we introduce some techniques.

- *Fixed-value:* A fixed value is considered as the number of substitutions for each activity such that for any $a \in act(EL')$, $NS_a = n$ where $n \in \mathbb{N}_{>1}$.

- *Selective:* By this technique only the sensitive frequencies are targeted to get perturbed. Hence, only some of the activities having the sensitive frequencies are decomposed. Here, we allocate the substitutions such that for any $a \in act(EL')$: $NS_a = \lceil {\#_a(EL')}/{\#_{median}(EL')} \rceil$ if $\#_a(EL') = \#_{most}(EL')$, and for any $a \in act(EL')$: $NS_a = \lceil {\#_a(EL')}/{\#_{least}(EL')} \rceil$ if $\#_a(EL') \in sstv_{lower}(EL') \setminus \#_{least}(EL')$. Note that we aim to perturb the bounds of frequencies with the minimum number of activities after the decomposition.
- *Frequency-based:* The substitutions are allocated based on the relative frequencies of the main activities. Here, we allocate the substitutions in such a way that for any $a \in act(EL')$, $NS_a = \lceil {\#_a(EL')}/{\#_{sum}(EL')} \times 100 \rceil$.

After specifying the number of substitutions for activity a, we make a substitution set $Sub_a = \{sa_1, sa_2, ..., sa_{NS_a}\}$ such that for any $a_1, a_2 \in act(EL')$: $Sub_{a_1} \cap Sub_{a_2} = \emptyset$ if $a_1 \neq a_2$.[3] Note that $Decom(act(EL')) = \{sa \in D(H(\mathcal{A})) \mid \exists_{a \in act(EL')} sa \in Sub_a\}$. To preserve the main feature of the vectors, we distribute the frequency of the main activity uniformly among its substitutions. To this end, while going through the event log, for each resource, the i^{th} occurrence of the activity $a \in act(EL')$ is replaced by the $sa_i \in Sub_a$, and when $i > NS_a$, i is reset to 1 (*round-robin* manner). Thereby, we guarantee that if the frequency of performing an activity by a resource is greater than or equal to the other resources, the frequency of performing the corresponding substitutions will also be greater or equal to the others.[4]

5.2 Privacy Analysis

To analyze the privacy, we measure the disclosure risk of the original event log, and the sanitized event logs. Two factors are considered to measure the disclosure risk including; the *number of activities* having the sensitive frequencies, and the *presence* of the actual activities having the sensitive frequencies. The presence for each bound of the frequencies before applying the decomposition method is $prs_{bound}(EL) = 1$ if $sstv_{bound}(EL) \neq \emptyset$. Otherwise, $prs_{bound}(EL) = 0$. For the sanitized event logs the presence is obtained as follows.

$$prs_{bound}(EL'') = \frac{|act(sstv_{bound}(EL'')) \cap \{sa \in Decom(act(EL')) \mid \#_a(EL') \in sstv_{bound}(EL')\}|}{|\{sa \in Decom(act(EL')) \mid \#_a(EL') \in sstv_{bound}(EL')\}|}$$

Also for each bound of the frequencies, $PR_{bound}(EL) = {1}/{|act(sstv_{bound}(EL))|}$ is the raw probability of activity disclosure based on the number of activities having the sensitive frequencies, and $DR_{bound}(EL) = {prs_{bound}(EL)}/{|act(sstv_{bound}(EL))|}$ is the disclosure risk. The whole disclosure risk w.r.t the assumed background knowledge is measured as follows.

$$DR(EL) = \frac{\alpha \times prs_{upper}(EL)}{|act(sstv_{upper}(EL))|} + \frac{(1-\alpha) \times prs_{lower}(EL)}{|act(sstv_{lower}(EL))|}$$

If $prs_{bound}(EL) = 0$ or $|act(sstv_{upper}(EL))| = 0$, $DR_{bound}(EL) = 0$. Also, α is utilized to set the importance of each bound of the frequencies.

[3] Note that the substitution sets should not be revealed.

[4] We consider a dummy resource in case there is an activity without resource.

Table 4: Similarity between JSN and JSN'' for the *fixed-value* technique

Threshold	Dataset	NS = 2		NS = 4		NS = 8		NS = 16	
		CN	UC	CN	UC	CN	UC	CN	UC
$\Theta = 0.1$	BPIC 2012	1.0	1.0	1.0	1.0	0.99	1.0	0.99	1.0
	BPIC 2017	1.0	1.0	1.0	1.0	0.99	1.0	0.98	1.0
$\Theta = 0.2$	BPIC 2012	1.0	1.0	0.99	1.0	0.98	1.0	0.95	1.0
	BPIC 2017	1.0	1.0	1.0	1.0	0.99	1.0	0.97	1.0
$\Theta = 0.3$	BPIC 2012	1.0	1.0	0.98	1.0	0.95	1.0	0.90	1.0
	BPIC 2017	1.0	1.0	1.0	1.0	0.97	1.0	0.95	1.0
$\Theta = 0.4$	BPIC 2012	1.0	1.0	0.97	1.0	0.92	1.0	0.88	1.0
	BPIC 2017	1.0	1.0	0.99	1.0	0.97	1.0	0.93	1.0
$\Theta = 0.5$	BPIC 2012	1.0	1.0	0.94	1.0	0.91	1.0	0.87	1.0
	BPIC 2017	1.0	1.0	0.99	1.0	0.96	1.0	0.93	1.0
$\Theta = 0.6$	BPIC 2012	1.0	1.0	0.94	1.0	0.90	1.0	0.85	1.0
	BPIC 2017	1.0	1.0	0.98	1.0	0.95	1.0	0.94	1.0
$\Theta = 0.7$	BPIC 2012	1.0	1.0	0.95	1.0	0.91	1.0	0.87	1.0
	BPIC 2017	1.0	1.0	0.99	1.0	0.97	1.0	0.96	1.0
$\Theta = 0.8$	BPIC 2012	1.0	1.0	0.96	1.0	0.95	1.0	0.93	1.0
	BPIC 2017	1.0	1.0	0.99	1.0	0.98	1.0	0.93	1.0
$\Theta = 0.9$	BPIC 2012	1.0	1.0	0.99	1.0	0.96	1.0	0.95	1.0
	BPIC 2017	1.0	1.0	0.99	1.0	0.96	1.0	0.92	1.0
Average	BPIC 2012	1.0	1.0	0.96	1.0	0.94	1.0	0.91	1.0
	BPIC 2017	1.0	1.0	0.99	1.0	0.97	1.0	0.94	1.0
Total Average	BPIC 2012	1.0		0.98		0.97		0.955	
	BPIC 2017	1.0		0.995		0.985		0.97	

6 Evaluation

To evaluate our approach, we show the effect on the accuracy and privacy for two real life event logs (BPIC 2012 and 2017). To this end, we have implemented an interactive environment in Python. Figure 1 shows an output of our tool.[5]

6.1 Accuracy

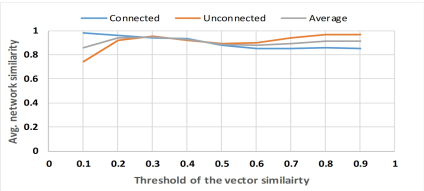
To examine the accuracy of our approach, we measure the similarity of joint -activities social networks from the original event log (JSN) and the sanitized event log (JSN''). To this end, we compare the similarity of their *connected (CN)* and *unconnected (UC)* parts. Note that $JSN = (res(EL), E)$, $JSN'' = (res(EL''), E'')$, and $res(EL) = res(EL'')$. Here, we use *Pearson* as the measure of similarity between vectors, which is one of the best measures according to [4].

$$CN = \frac{|E \cap E''|}{|E|} \qquad UC = \frac{|(res(EL) \times res(EL) \backslash E) \cap (res(EL) \times res(EL) \backslash E'')|}{|res(EL) \times res(EL) \backslash E|}$$

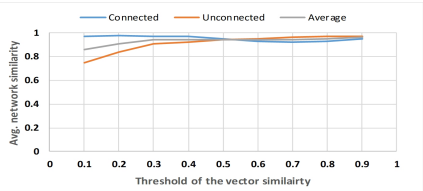
Table 4 shows the similarities when the *fixed-value* technique is used to identify the number of substitutions. As can be seen, the networks are almost the same and the accuracy is acceptable. When the number of substitutions increases, the average of similarities decreases, showing the typical trade-off between accuracy and privacy. Moreover, the networks in the unconnected parts are identical, i.e., if two resources are not connected in the JSN, there are not connected in the JSN'' as well.

Figure 2 shows the similarities w.r.t. various thresholds when using the *selective* or *frequency-based* technique. As can be seen, on average the *selective* technique leads to more accurate results. However, in the unconnected parts the *frequency-based* technique has better results. Note that BPIC 2017 is larger than BPIC 2012 in terms of both resources and activities (Table 5).

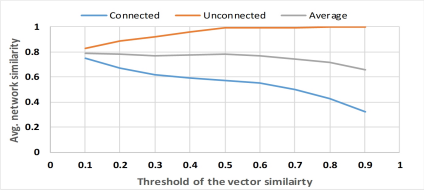
[5] https://github.com/m4jidRafiei/privacyAware-roleMining



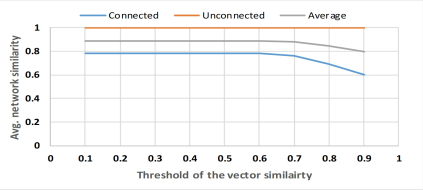
(a) Selective technique for BPIC 2012



(b) Selective technique for BPIC 2017



(c) Frequency-based technique for BPIC 2012



(d) Frequency-based technique for BPIC 2017

Fig. 2: The similarities between JSN and JSN'' when using the *selective* or *frequency-based* technique to identify the number of substitutions.

6.2 Privacy

To evaluate the effect on privacy, we calculate the disclosure risk on the original event logs and the sanitized event logs after applying the decomposition method with different techniques. Table 6 and Table 7 show the parameters regarding the disclosure risk for BPIC 2012 and 2017 respectively. As can be seen, when the *fixed-value* technique is used, DR is lower for the larger values as the number of substitutions in both event logs. Moreover, since the relative frequency of the least frequent activities is very low, the *frequency-based* technique does not affect the lower bound of sensitive frequencies. This weakness can be mitigated by combining this technique with the *fixed-value* such that the number of substitutions would be the relative frequency plus a fixed value.

To compare the introduced techniques, we consider the minimal disclosure risk which can be supplied by all the techniques as the basis of comparison and evaluate the *accuracy* and *complexity* provided by the different techniques for the same disclosure risk. The accuracy is the average similarity between the

Table 5: Statistics regarding frequencies in BPIC 2012 and BPIC 2017

	BPIC 2012	BPIC 2017
No. resources	69	145
No. unique activities	24	26
No. activities	262200	1202267
$\|upper(EL))\|$	5	5
Frequency of the most frequent activities ($\#_{most}(EL)$)	54850	209496
Relative frequency for any a: $\#_a(EL) \in \#_{most}(EL)$	0.20	0.17
$\|lower(EL))\|$	4	6
Frequency of the least frequent activities ($\#_{least}(EL)$)	12	22
Relative frequency for any a: $\#_a(EL) \in \#_{least}(EL)$	4×10^{-5}	1×10^{-5}

Table 6: The DRs before and after applying the method on BPIC 2012

	PR_{upper}	PR_{lower}	prs_{upper}	prs_{lower}	$DR(\alpha = 0.5)$
BPIC 2012	0.5	0	1	0	0.25
Fixed-value NS=2	0.25	0	1	0	0.12
Fixed-value NS=4	0.25	0	0.5	0	0.06
Fixed-value NS=8	0.12	0	0.5	0	0.03
Fixed-value NS=16	0.06	0	0.5	0	0.01
Selective	1	0	0.09	0	0.04
Frequency-based	0.5	0	0.04	0	0.01

Table 7: The DRs before and after applying the method on BPIC 2017

	PR_{upper}	PR_{lower}	prs_{upper}	prs_{lower}	$DR(\alpha = 0.5)$
BPIC 2017	0.25	0.5	1	1	0.37
Fixed-value NS=2	0.5	0.25	0.25	1	0.18
Fixed-value NS=4	0.25	0.12	0.25	1	0.09
Fixed-value NS=8	0.12	0.07	0.25	1	0.05
Fixed-value NS=16	0.06	0.04	0.25	1	0.03
Selective	1	0.2	0.09	0.41	0.08
Frequency-based	0.33	0.5	0.04	1	0.25

networks, and the complexity is considered as the number of unique activities. Note that for the *fixed-value* technique, we inspect the event log which has the minimum NS providing the basis disclosure risk. Table 8 and Table 9 show the results of this experiment for BPIC 2012 and 2017 respectively. As one can see, in both event logs, the *fixed-value* technique provides more accurate results and the *selective* technique imposes less complexity.

All the above-mentioned explanations and our experiments demonstrate that the decomposition method provides accurate and highly flexible protection for mining roles from event logs, e.g., the decomposition method with the *frequency-based* technique can be used when the upper bound of frequencies is more sensitive and the accuracy of the unconnected parts is more important.

Table 8: Comparison of techniques in BPIC 2012

	DR ($\alpha = 0.5$)	Accuracy	Complexity
Fixed_value NS=8	0.04	0.97	188
Selective	0.04	0.9	87
Frequency-based	0.04	0.75	108

Table 9: Comparison of techniques in BPIC 2017

	DR ($\alpha = 0.5$)	Accuracy	Complexity
Fixed_value NS=2	0.25	1	52
Selective	0.25	0.93	43
Frequency-based	0.25	0.87	113

7 Conclusions

In this paper, for the first time, we focused on privacy issues in the organizational perspective of process mining. We proposed an approach for discovering joint-activities social networks and mining roles w.r.t. privacy. We introduced the *decomposition* method along with a collection of techniques by which the private information about the individuals would be protected against frequency-based attacks. The discovered roles can be replaced with individuals in the event data for further performance and bottleneck analyses.

The approach was evaluated on BPIC 2012 and 2017, and the effects on accuracy and privacy were demonstrated. To evaluate the accuracy, we measured the similarity between the connected and unconnected parts of two networks separately while different thresholds were considered. Moreover, we introduced three different techniques to identify the number of substitutions in the decomposition method, and we showed their effect on the accuracy and privacy, when the frequencies of activities are assumed as background knowledge. In the future, other techniques or combination of the introduced ones could be explored with respect to the characteristics of the event logs.

References

1. van der Aalst, W.M.P.: Process mining: data science in action. Springer (2016)
2. van der Aalst, W.M.P.: Responsible data science: using event data in a "people friendly" manner. In: International Conference on Enterprise Information Systems. pp. 3–28. Springer (2016)
3. van der Aalst, W.M.P., Adriansyah, A., De Medeiros, A.K.A., Arcieri, F., Baier, T., Blickle, T., Bose, J.C., Van Den Brand, P., Brandtjen, R., Buijs, J., et al.: Process mining manifesto. In: International Conference on Business Process Management. pp. 169–194. Springer (2011)
4. van der Aalst, W.M.P., Reijers, H.A., Song, M.: Discovering social networks from event logs. Computer Supported Cooperative Work (CSCW) **14**(6), 549–593 (2005)
5. Agrawal, R., Srikant, R.: Privacy-preserving data mining, vol. 29. ACM (2000)
6. Burattin, A., Conti, M., Turato, D.: Toward an anonymous process mining. In: Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on. pp. 58–63. IEEE (2015)
7. Fahrenkrog-Petersen, S.A., van der Aa, H., Weidlich, M.: Pretsa: Event log sanitization for privacy-aware process discovery. In: 1st IEEE International Conference on Process Mining (2019)
8. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: 2007 IEEE 23rd International Conference on Data Engineering. pp. 106–115. IEEE (2007)
9. Mannhardt, F., Petersen, S.A., Oliveira, M.F.: Privacy challenges for process mining in human-centered industrial environments. In: 2018 14th International Conference on Intelligent Environments (IE). pp. 64–71. IEEE (2018)
10. Rafiei, M., von Waldthausen, L., van der Aalst, W.M.P.: Ensuring confidentiality in process mining. In: Proceedings of the 8th International Symposium on Data-driven Process Discovery and Analysis (SIMPDA 2018), Seville, Spain, December 13-14, 2018. pp. 3–17 (2018)
11. Tillem, G., Erkin, Z., Lagendijk, R.L.: Privacy-preserving alpha algorithm for software analysis. In: 37th WIC Symposium on Information Theory in the Benelux/6th WIC/IEEE SP