PMDG: Privacy for Multi-Perspective Process Mining through Data Generalization

Ryan Hildebrant[1], Stephan A. Fahrenkrog-Petersen[2,3], Matthias Weidlich[2], and Shangping Ren[4]

[1] University of California, Irvine, rhildebr@uci.edu
[2] Humboldt-Universität zu Berlin, {fahrenks,weidlima}@hu-berlin.de
[3] Weizenbaum Institute for the Networked Society
[4] San Diego State University, sren@sdsu.edu

Abstract. Anonymization of event logs facilitates process mining while protecting sensitive information of process stakeholders. Existing techniques, however, focus on the privatization of the control-flow. Other process perspectives, such as roles, resources, and objects are neglected or subject to randomization, which breaks the dependencies between the perspectives. Hence, existing techniques are not suited for advanced process mining tasks, e.g., social network mining or predictive monitoring. To address this gap, we propose PMDG, a framework to ensure privacy for multi-perspective process mining through data generalization. It provides group-based privacy guarantees for an event log, while preserving the characteristic dependencies between the control-flow and further process perspectives. Unlike existing privatization techniques that rely on data suppression or noise insertion, PMDG adopts data generalization: a technique where the activities and attribute values referenced in events are generalized into more abstract ones, to obtain equivalence classes that are sufficiently large from a privacy point of view. We demonstrate empirically that PMDG outperforms state-of-the-art anonymization techniques, when mining handovers and predicting outcomes.

Keywords: Privatization · K-anonymity · Attribute Generalization

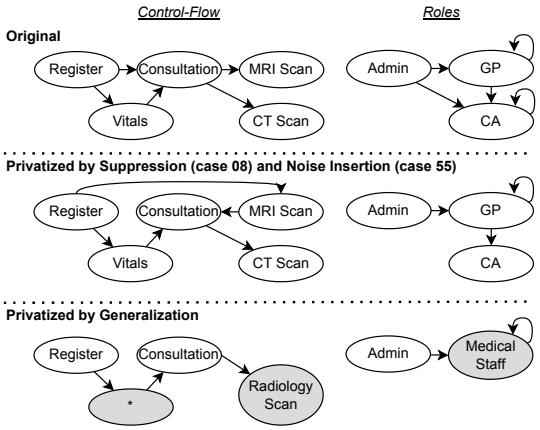
1 Introduction

Privacy-preserving process mining [1] enables data-driven analysis of business processes, while protecting sensitive data about the individuals involved in process execution. To this end, existing techniques rely on the anonymization of an event log, which is commonly modeled as a set of traces, with each trace being a sequence of events that denote activity executions. In order to obtain a provable privacy guarantee, the traces of an event log are transformed. Here, existing techniques differ in terms of the adopted privacy guarantee and the properties preserved by these transformations. Anonymization of event logs may guarantee differential privacy [2] or rely on group-based notions, such as k-anonymity and its derivatives [3]. Moreover, the respective transformations may only suppress behavior in the log or potentially introduce new and noisy behavior in terms of unseen sequences of activity executions.

Most techniques for privacy-preserving process mining [4–6] focus on the construction of a process model from an event log [7]. As such, they target the control-flow per-

Case	Activity	Location	Role
07	Register	Day Clinic	Admin
07	Vitals	Day Clinic	GP
07	Consultation	Day Clinic	GP
07	CT Scan	Hospital	CA
...	...	...	...
08	Register	Hospital	Admin
08	Consultation	Hospital	CA
08	MRI Scan	Hospital	CA
...	...	...	...
Noise trace added by privatization:			
55	Register	Hospital	Admin
55	MRI Scan	Day Clinic	GP
55	Consultation	Day Clinic	Admin
...	...	...	...

(a)



(b)

Fig. 1: (a) Example log of a clinical pathway containing traces for case 07 and 08; (b) the control-flow and the role perspective, when considering the original log, the event log privatized by suppression (case 08 is suppressed) and noise insertion (case 55 is added), and the event log privatized by generalization of activities and roles.

spective of the process, trying to ensure that the anonymized event log includes semantically correct sequences of activity executions from the process. However, event logs also contain information on other process perspectives, such as roles, resources, and case objects. Advanced process mining tasks exploit the relation between the control-flow and these additional perspectives, e.g., to extract hand-overs during process execution [8] or to construct models to predict the outcome of running process instances [9].

As of today, ensuring privacy beyond the control-flow creates a notable research gap. So far, data linked to events is either neglected, or assigned randomly once the control-flow has been anonymized [10]. The latter tends to break any dependencies between the various perspectives, rendering the event logs unsuitable for multi-perspective process mining, as illustrated in Fig. 1. Here, Fig. 1a shows an example event log of a clinical pathway for two patients, cases 07 and 08. Fig. 1b (top), in turn, highlights the control-flow dependencies and the hand-overs between the involved roles. Existing techniques for privatization of the control-flow by suppression (of case 08) and noise insertion (of case 55), however, do not only disturb the control-flow, but also break the dependencies between process perspectives, as illustrated for the hand-overs between roles in Fig. 1b (middle). This raises the question of *how to preserve the characteristic dependencies between the process perspectives when anonymizing an event log*.

In this paper, we address this question with PMDG, a framework to ensure privacy for multi-perspective process mining through data generalization. It preserves the dependencies between process perspectives when constructing an event log that meets k-anonymity, a privacy guarantee often adopted in industry [11]. To this end, we adopt data generalization instead of data suppression and noise insertion. Fig. 1b (bottom) gives the intuition of this approach: here, sufficiently large equivalence classes of traces are derived by generalizing activities (*CT Scan* and *MRI Scan* become *Radiology Scan*) and role information (*GP* and *CA* become *Medical Staff*). While the generalization incurs some information loss, it arguably preserves general characteristics, such as the

conduct of radiologic scans only after the consultation, as well as the handovers between administrative staff and medical personnel.

In sum, our contributions are the definition of PMDG as a first framework to enable privacy for multi-perspective process mining; and its instantiation with specific techniques for (i) the vectorization of traces to facilitate control-flow generalization; (ii) the selection of hierarchies to be used for the abstraction; and (iii) the application of the selected hierarchies to generalize the control-flow and the data assigned to events.

We demonstrate the effectiveness of PMDG for multi-perspective process mining in experiments with three public event logs. When mining handovers and predicting process outcomes, we observe that PMDG significantly outperforms state-of-the-art anonymization techniques in terms of maintaining characteristic hand-overs and classification accuracy, respectively. In the remainder, Section 2 reviews related work on privacy-preserving process mining. Section 3 then provides background information. The PMDG framework is outlined and instantiated in Section 4, before we present evaluation experiments in Section 5. We discuss our approach on a qualitative level in Section 6, before concluding in Section 7.

2 Related Work

Privacy-preserving process mining has received much attention recently [1]. Several approaches have been proposed to ensure k-anonymity and other group-based privacy guarantees, e.g., by merging similar traces [4,12] or filtering data [5]. Due to their focus on the control-flow, these methods are not suited for multi-perspective process mining.

Instead of hiding sensitive data within groups of traces, some approaches achieve differential privacy by inserting noise into event logs [6,13,14]. Here, the privacy guarantees limits the effect one individual can have on the anonymized data. Yet, the approaches filter behavior from the log or introduce new and formerly unseen behavior.

The importance of the privatization of additional process perspectives has been highlighted in [15], which introduced a technique that is tailored to one particular perspective, i.e. resource assignments. In the general case, the aforementioned approaches for control-flow anonymization may be combined with an enrichment step, which either assigns values randomly [10] or unifies their distribution over an event log [16]. Either way, characteristic dependencies between the process perspectives are compromised and the insertion of new dependencies may lead to wrong conclusions in the analysis.

Another angle is followed in confidential process mining that aims to protect an event log by encryption [17]. This may include several process perspectives, but lacks any formal guarantee on the privacy of individuals in the dataset.

Our approach relies on data generalization to privatize several process perspectives. Generalization is a well-established method to privatize relational, hierarchical, and simple sequence data, see [18–20]. Yet, PMDG is the first use of data generalization for event logs, i.e. multi-variate sequences for which the dependencies between the various dimensions shall be preserved.

3 Background

Below, we summarize common notions for event logs and group-based privacy guarantees, as they are required for the definition of PMDG.

Event Logs. Process mining is based on events, each representing the recorded execution of an activity, i.e. *Register* or *MRI Scan* in Fig. 1. We denote the universes of activities and events by $\mathcal{A}$ and $\mathcal{E}$, respectively. The activity for which an event $e \in \mathcal{E}$ signals the execution is written as $e.a \in \mathcal{A}$. Events have a schema, defined by a set of attributes, $\mathcal{D} = \{D_1, \ldots, D_n\}$, and we denote the domain of values of attribute D by $\mathcal{V}_D$. For an event e, we write $e.D \in \mathcal{V}_D$ for the respective value of attribute D. In Fig. 1, we have $\mathcal{D} = \{Location, Role\}$. Here, attribute $Role$ assumes the values $Admin$, CA, and GP, whereas the respective domain may also include further values. In particular, it can contain more abstract roles, such as $Medical\ Staff$ or $Staff$.

Events that relate to the same, single execution of a process are grouped into a trace. Each trace σ is a finite sequence of events $\langle e_1, \ldots, e_n \rangle \in \mathcal{E}^*$ of length $|\sigma| = n$. We use $\sigma.A$ to denote the control-flow of a trace σ, meaning the sequence of activities for which the execution is indicated by the events within the trace. For our running example, for instance, we have $\sigma.A = \langle Register, Vitals, Consultation, CT\ Scan \rangle$ for the trace of case 07. All traces with the same control-flow are said to be of the same trace variant, which is identified by one of the respective traces. That is, $[\![\sigma]\!]^A$ is the bag of traces that have the same control-flow as σ. A bag of traces is called an event log, $L = [\sigma_1, \ldots, \sigma_n]$. It represents the input for many process mining algorithms.

k-Anonymity. A well-known way to protect the privacy of individuals is to hide them within a group, which is the aim of the k-anonymity privacy guarantee [3]. The idea is that, in a dataset (an event log in our setting), one individual shall be indistinguishable from at least $k - 1$ other individuals. Therefore, the probability of identifying one individual, the so-called problem of *identity disclosure*, can be bound to $1/k$. To achieve that k individuals are indistinguishable, the quasi-identifiers need to be aligned. In general, quasi-identifiers are attributes that enable the identification of an individual, such as a postcode or birth date.

In our setting, we assume that all attributes of all events and the control-flow of a trace can serve as quasi-identifiers. We therefore consider all traces that have the same control-flow and sequence of selected attribute values to be part of an equivalence class. The selected attributes are generated based on a defined perspective required for advanced process mining tasks. In line with the notation introduced for trace variants, we identify an equivalence class by one of its members, i.e. $[\![\sigma \mid \mathcal{D}']\!]$ denotes an equivalence class that comprises all traces that have the same control-flow and attribute values for the specified attributes $\mathcal{D}' \subseteq \mathcal{D}$ as σ. Based thereon, we define k-anonymity as follows: Let $L = [\sigma_1, \ldots, \sigma_n]$ be an event log. Then, the log L satisfies *k-anonymity* with respect to a given perspective, if for every equivalence class $[\![\sigma \mid \mathcal{D}']\!]$ in L induced by a trace $\sigma \in L$, it holds that $|[\![\sigma \mid \mathcal{D}']\!]| \geq k$.

We note that the above definition of equivalence classes, and hence of k-anonymity, induces the strictest possible notion. It assumes the strongest adversary, under which any attribute and the control-flow may serve as quasi-identifiers in an identity disclosure attack. As such, it subsumes attacks in which an adversary possesses only a certain type

of background knowledge [5], such as knowing which activities have been executed by an individual, but not the order of their execution. In order to avoid assumptions on the background knowledge of an adversary, we adopt the above model that represents the worst case scenario.

4 Generalization of Event Logs

This section first outlines the design principles for our work (Section 4.1). Then, we give an overview of our PMDG framework to address the identified research gap (Section 4.2). While some steps of it rely on existing techniques, some aspects call for new techniques to ensure high utility of the anonymized event log. Specifically, we introduce strategies for trace vectorization (Section 4.3) and hierarchy selection (Section 4.4).

4.1 Design Principles of the Framework

We developed the PMDG framework using the design science methodology [21]. The starting point for our problem observation is that existing anonymization techniques for event logs mostly rely on noise insertion and aggregation, but do not incorporate any generalization strategies. To address this research gap, we derived the following design objectives for the artifact: Given an event log with m traces and a privacy parameter $k \leq m$, the artifact (i) shall transform the event log to fulfill k-anonymity through generalization; and (ii) shall minimize the total amount of generalizations that are applied to the log. Within the remainder of this section, we introduce the PMDG framework to realize these design objectives. The evaluation step of the artifact is provided within a later section, while this paper denotes the final communication step of the design science methodology.

4.2 PMDG Framework

As shown in Fig. 2, our framework is applied to an event log that contains information about multiple process perspectives through the attribute values assigned to events (see Section 3). In addition, it relies on generalization hierarchies. That is, an *activity hierarchy*, modeled as a function $\rho_A : \mathcal{A} \rightarrow \mathcal{A}' \cup \{\star\}$, which maps an activity to a more abstract activity or a wildcard $\star$. For an attribute D representing an an additional perspective, a *value hierarchy* $\rho_D : \mathcal{V}_D \rightarrow \mathcal{V}_D \cup \{\star\}$ maps an attribute value to a more abstract value or a wildcard. Either way, for the control-flow or perspective, the hierarchies are rooted in the wildcard $\star$, meaning that for any activity $a \in \mathcal{A}$ and value $v \in \mathcal{V}_D$, it holds that $\star \in \rho_A^*(a)$ and $\star \in \rho_D^*(v)$, where ρ^* denotes the transitive application of a generalization hierarchy ρ. Moreover, for all process perspectives, multiple hierarchies may be available to generalize the respective information. Using the hierarchies, the PMDG framework transforms the event log given as input, to one such that the resulting log guarantees k-anonymity.

Common strategies for data generalization are based on operations that change individual values of the elements in a dataset [22]. Hence, in order to enable comprehensive generalization, i.e. to achieve that any two elements may end up in the same equivalence

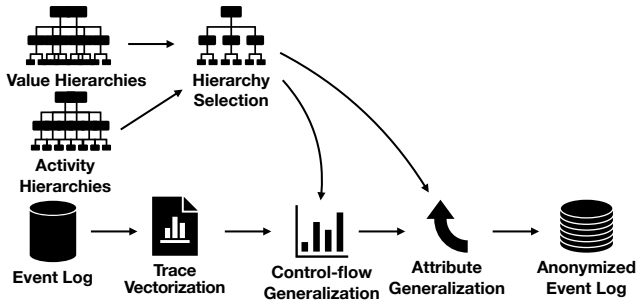


Fig. 2: Overview of the PMDG framework.

class, it is necessary to ensure that all elements assume the same structure. Transferred to our setting, this requires all traces to be of equal lengths and all events to have the same schema. While the latter requirement typically does not impose any challenges in practice and is incorporated in our definition of the model already, differences in the lengths of traces need to be handled. To this end, our framework incorporates *trace vectorization* as a first step, which we explain in more detail in Section 4.3.

Next, the PMDG framework generalizes the control-flow before considering further process perspectives. The reason being that behavioral information serves as the basis for advanced process mining tasks and shall always be protected by k-anonymity. Further perspectives enrich the control-flow and may be subject to more fine-granular control of the privacy guarantee, e.g., adopting t-closeness [23] instead of k-anonymity.

To obtain the privacy guarantee through generalization, however, multiple hierarchies may be available, for the control-flow as well as for other process perspectives. For instance, activities may be generalized according to the artifact that is handled (e.g., all activities related to a *CT Scan* are generalized into a single activity) or by the type of the action that is conducted (e.g., all activities that prescribe different drugs are combined into a single activity). Similarly, roles may be generalized based on some organizational structure (e.g., wards in a hospital) or some ability (e.g., the specialization of a doctor). Each hierarchy will affect the utility of the resulting event log differently. Hence, the PMDG framework includes a step to guide the selection of one hierarchy for the control-flow generalization, and one hierarchy for each attribute representing an additional process perspective, as detailed in Section 4.4.

Once the generalization hierarchies have been selected, they are applied to the event log using existing algorithms to achieve k-anonymity [22, 24]. In essence, these algorithms adopt some step-wise generalization until the resulting equivalence classes are sufficiently large. Specifically, in our context, given an activity hierarchy ρ_A and a value hierarchy ρ_D per attribute D, the result of the generalization step can be char-

Traces after trace vectorization:

Case 07	
Activity	Role
Register	Admin
Vitals	GP
Consultation	GP
CT Scan	CA

Case 08	
Activity	Role
Register	Admin
⋆	⋆
Consultation	CA
MRI Scan	CA

Traces after generalization:

Case 07	
Activity	Role
Register	Admin
⋆	⋆
Consultation	Medical Staff
Radiology Scan	CA

Case 08	
Activity	Role
Register	Admin
⋆	⋆
Consultation	Medical Staff
Radiology Scan	CA

Fig. 3: Generalization example.

acterized as follows: For each trace $\sigma = \langle e_1, \ldots, e_n \rangle$ of the original log L, the resulting log L' will contain a trace $\sigma' = \langle e'_1, \ldots, e'_n \rangle$, such that for each event e_j, $1 \leq j \leq n$, it holds that the activity and each attribute value remains unchanged or has been generalized, i.e. $e'_j.a = e_j.a$ or $e_j.a = \rho^*_A(e_j.a)$ and $e'_j.D = e_j.D$ or $e_j.D = \rho^*_D(e_j.D)$.

For example, consider the traces for cases 07 and 08 from Fig. 1 and focus solely on the control-flow and the role perspective. Trace vectorization will normalize the length of both traces by inserting an event with wildcard values (discussed in Section 4.3), see Fig. 3 (top). With an activity hierarchy that generalizes *MRI Scan* and *CT Scan* to *Radiology Scan*, as well as a value hierarchy generalizing *GP* and *CA* to *Medical Staff*, the traces can be generalized to fall into the same equivalence class, see Fig. 3 (bottom). As such, the resulting event log would satisfy 2-anonymity.

4.3 Trace Vectorization

As explained above, comprehensive generalization requires that all elements in a dataset assume the same structure. Furthermore, we want to ensure that an anonymized event log can be generated for every k that is equal or smaller than the number of the traces in the log. However, only traces with the same length can be merged into the same equivalence class. Therefore, we need to unify the length of traces in the event log. To this end, we adopt trace vectorization, which is similar to the idea of sequence encoding in predictive process monitoring [25]. Specifically, given an event log $L = [\sigma_1, \ldots, \sigma_n]$, trace vectorization yields a log $L' = [\sigma'_1, \ldots, \sigma'_n]$, such that:

- All traces have the same length, i.e. for all σ'_i, σ'_j in L', it holds $|\sigma'_i| = |\sigma'_j|$.
- For each trace $\sigma = \langle e_1, \ldots, e_m \rangle$ of the original log L, there is a corresponding trace $\sigma' = \langle e'_1, \ldots, e'_k \rangle$ in L', so that the projection of σ' on the events $\{e_1, \ldots, e_m\}$ yields the trace σ and all events e of σ' that are not part of the projection are wildcard elements, i.e., it holds that $e.A = \star$ and $e.D = \star$ for all $D \in \mathcal{D}$.

One naive approach for trace vectorization would be to extend all traces that are shorter than the maximum length of traces in an event log with wildcard events at the end. However, such an approach cannot be expected to preserve the utility of the traces for process mining, especially considering the control-flow perspective. For instance, for the trace of case 08 in Fig. 3, adding the wildcard event at the end would have severe consequences for the subsequent generalization: Instead of preserving the information on the *Consultation* and *Radiology Scan* activities of both traces, all except the first activity would be generalized to the root element ($\star$).

In PMDG, therefore, we employ a strategy based on multi-sequence alignments (MSA) [26]. In essence, MSA identifies how to insert gaps into sequences of symbols, such that the same symbol is assigned to a certain index in all sequences and the number of gap indices is minimal. In our setting, we adopt MSA for the control-flow perspective, as it serves as the basis for process mining tasks. That is, given an event log $L = [\sigma_1, \ldots, \sigma_n]$, MSA is applied to the set $\{\sigma_1.A, \ldots, \sigma_n.A\}$ to identify where wildcard events shall be inserted.

4.4 Hierarchy Selection

As detailed above, multiple hierarchies may be employed to generalize the control-flow or the data representing additional process perspectives. Below, we first elaborate on types of hierarchies and their origin as well as their implications for the utility of the anonymized event log. We then present a heuristic solution to guide the selection of generalization hierarchies as part of PMDG.

Types of hierarchies. In general, one can distinguish two types of hierarchies:

(i) *Syntactic hierarchies* are obtained by suppressing a part of an activity label or an attribute value. Common examples for syntactic hierarchies are numeric values (e.g., postcodes *'12489'* and *'12555'* are generalized to *'12—'*) or dates (*'10/2022'* and *'12/2022'* are generalized to *'–/2022'*). However, one may also consider activities and generalize, for instance, *CT Scan* and *MRI Scan* to *Scan* by suppressing the first token of the label.

(ii) *Semantic hierarchies* generalize the meaning of an activity or attribute value. An example would be the generalization of an attribute capturing a city (*'Berlin'*) into a country (*'Germany'*), larger region (*'EU'*), or continent (*'Europe'*). The creation of semantic hierarchies requires domain knowledge and these hierarchies are usually either user-defined or extracted from a knowledge base. For activities in traditional business processes, for instance, the MIT process handbook [27] defines generalization hierarchies of activities.

The selection of a hierarchy will impact the utility of the resulting event log, even when considering only a single type of hierarchy. Taking up the example of syntactic generalizations of dates, *'11.2022'* and *'12.2022'* may be generalized not only to *'–/2022'*, but also to *'11/—'* and *'12/—'*, respectively, depending on which parts to suppress. Either generalization provides a different kind of information, which influences the types of questions that can be answered with process mining for the anonymized log.

Selecting a hierarchy. Since the selection of certain hierarchies for data generalization has significant implications, ideally, one would test all available hierarchies for the control-flow and all attributes. Measuring the quality of the resulting event logs based on a chosen utility measure, the best combination of hierarchies can be determined. However, such a brute-force approach is typically infeasible, due to the exponential number of hierarchy combinations. Therefore, in PMDG, we incorporate a heuristic strategy to guide the selection of a generalization hierarchy independently for the control-flow and each attribute. The heuristic is based on a notion of utility, for which we consider the following instantiations:

- The utility is given by the number of equivalence classes within an anonymized event log. Here, the intuition is that a larger number of equivalence classes in the anonymized log yields a better representation of the variance in the original log.
- The utility is inversely proportional to the differences in size of the equivalence classes, i.e. the number of contained traces. Here, the motivation is to preserve information on common behavior more precisely than on uncommon behavior.

Based on a specific notion of utility, the selection of a hierarchy per process perspective may be guided by an estimated utility, as follows. Let $\{\rho_D^1, \ldots, \rho_D^n\}$ be a set of hierarchies for an attribute D (or, analogously, for the activities). Then, for each hierarchy, we determine the equivalence classes when considering *only* the attribute D

and *one* level in the generalization hierarchy (i.e., $\rho_D^1, \dots \rho_D^n$ are applied only once, not transitively). Let u_1^i be the utility as determined for the equivalence classes obtained with ρ_D^i, which, as mentioned above, may be defined by the number of classes. Afterwards, the equivalence classes obtained with subsequent levels of the hierarchies are assessed iteratively, yielding utility values u_j^i for hierarchy ρ_D^i when incorporating it up to level j. Per hierarchy ρ_D^i, these utility values are summed up in a weighted manner, i.e., $u^i = \sum_{j=1}^{k} w_j \cdot u_j^i$ with k as the maximum depth of the hierarchy. Using the weights w_j enables us to give preference to different levels of generalization, i.e., prioritizing the generalization from a city to a country, over the one from a country to a continent. Finally, we select a hierarchy ρ_D^i for which the estimated utility u^i is maximal over all hierarchies $\{\rho_D^1, \dots \rho_D^n\}$ for attribute D.

5 Evaluation

Within this section, we investigate how anonymizing event logs with PMDG impacts the utility of advanced process mining tasks. Through an empirical evaluation, we show the feasibility and effectiveness of PMDG. First, we will give an overview of the datasets used in our experiments in Section 5.1. Next, we outline our experimental setup, baseline, and evaluation metrics in Section 5.2. Finally, we present the results of our experiments in Section 5.3.

5.1 Datasets and Implementation

For our experiments, we use three real-world event logs: BPIC 2013 [28], Road Traffic Fines [29], and the CoSeLoG [30]. For each log, we excluded all variants that only appear once. This ensures a reasonable setting for anonymization (where unique traces would be problematic in any case). Certain experiments with advanced process mining tasks required the existence of the same attribute in all events, in that case we performed these experiments only with the BPIC 2013 [28] and CoSeLoG [30] event logs, since road traffic fines is missing such an attribute.

For all of our experiments, we provide an open-source implementation on GitHub.[5] The trace generalization approach is implemented in Python. For the generalizations of attributes, we used Java libraries from the ARX project.[6] For our experiments on mining handovers, we relied on the organizational mining features of PM4Py.[7] For our experiments on outcome prediction, we used scikit-learn.[8]

5.2 Experimental Setup

Parameter settings. In our experiments, we use different strengths for k-anonymity, with values of k varying from $\{5, 10, 15, 20\}$. Furthermore, we use semantic hierarchies that we created manually. We also tested syntactic hierarchies, but these were

[5] https://github.com/Ryanhilde/PMDG_Framework/
[6] https://arx.deidentifier.org
[7] https://pm4py.fit.fraunhofer.de
[8] https://scikit-learn.org

Table 1: Comparison of Control-flow Preservation

Log	Trace Vec.	$k = 5$	$k = 10$	$k = 15$	$k = 20$
CoSeLoG	MSA	**17**	**13**	**9**	**8**
	Naive	**17**	**13**	8	**8**
BPIC 2013	MSA	**82**	31	23	**23**
	Naive	65	**34**	**27**	22
Traffic Fines	MSA	**75**	**53**	**43**	**37**
	Naive	12	12	12	12

always outperformed in terms of the amount of changes applied to the anonymized log. We selected our semantic hierarchies based on retained equivalence classes for the control-flow and minimum generalizations for the attributes. In our experiments regarding predictive process monitoring, we trained decision trees on 1,000 randomly generated 20/80 test-train splits.

Baseline. As a baseline for some of our experiments, we used PRIPEL [10], a framework that transforms event logs to achieve ϵ-differential privacy [2]. The provided privacy guarantee is not directly comparable with k-anonymity, i.e. we cannot assume that a specific k-value will ensure the same amount of privacy as a setting in PRIPEL. However, PRIPEL is the best choice for comparison, as it is the only existing technique that is capable of handling all attribute values. In general, a lower value for ϵ corresponds to a stronger privacy guarantee. For our experiments, we consider two settings for PRIPEL in the two event logs, a weak privacy guarantee ($\epsilon = 1.0$) and a strong one ($\epsilon = 0.1$). Furthermore, we set the pruning parameter of PRIPEL to 2 in the weaker setting and to 20 for the stronger one; and always set the maximum prefix-length to the mean of the trace variants. These two parameters are required by PRIPEL, due to the underlying control-flow anonymization technique [6]. Furthermore, we compare our trace vectorization technique based on MSA with a naive approach, that fills up all traces at the end with wildcards.

Evaluation metrics. We use the number of remaining variants as a metric, to measure the control-flow preservation after the anonymization, which is shown in Table 1. To study the impact of the attribute anonymization on the utility of advanced process mining tasks, we investigated two advanced process mining techniques: the discovery of handovers [8, 31] and process outcome prediction [9].

Through handover analysis, an analyst can investigate which attribute values directly follow each other within two events of the same trace. Often this analysis is performed on resource related attributes such as resource role or location. In order to quantify the results of our anonymization, we measure the preserved information of the generalized event logs compared to the original event log. We utilize an information preservation metric to capture the information loss due to generalization. Our metric is based on the intuition that generalizing a handover from its original relation (e.g. in the case of resource locations: Germany to China) to a generalized relation (e.g. Europe to China) still has some utility. Furthermore, this utility is higher than if the handover would have been generalized to an even higher level (Europe to Asia) or the highest

level (Europe to World). We therefore define the preservation for generalized handovers p as:

$$p = \frac{[1 - \frac{\alpha(e'_1.D)}{\alpha(*)} + \frac{\alpha(e_1.D)}{|\alpha(*)}] + [1 - \frac{\alpha(e'_2.D)}{\alpha(*)} + \frac{\alpha(e_2.D)}{\alpha(*)}]}{2} \quad (1)$$

The assumption here is that α is a function that returns the number of potential attribute values that can be represented through a (generalized) attribute value, i.e. the value *'EU'* could represent 27 countries in an attribute D that encodes countries. The special case $\alpha(*)$ returns the number of fine-granular values of an attribute D, i.e. all possible countries. The values e_1 and e'_1 represent the original and generalized values for the left-side of the handover, respectively, while e_2 and e'_2 represent the right-side, i.e. (China). Our metric only measures the loss of information for existing handovers, since generalization cannot insert new handover relations.

As a second analysis task, we consider process outcome prediction. Here, the utility of an event log is given by the classification results. We assess these results using the well-known classification metrics: *precision*, the fraction of positively labeled instances that are actually correct; *recall*, the ratio of actual positives that are correctly labeled; and *F1-score*, the harmonic mean of precision and recall.

5.3 Results

Control-flow Preservation. In Table 1, we show that the MSA based trace vectorization usually outperforms the naive approach. We can see, it can provide significant benefits based on the traffic fines event logs. In cases where the naive approach is better, the benefit is comparatively small. Overall, we can observe that higher k lead to a higher loss in control-flow variance.

Handovers. In Fig. 4, we visualize the handovers created from the anonymized BPIC 2013 log based on the attribute *org:role* and $k = 5$. Such an analysis would allow an organization to understand which kind of resource roles usually interact with each other. We can clearly see that the anonymization through PMDG produces a smaller handover graph (middle graph) that contains less information as compared to the original handover graph (left graph). However, more detailed insights can be derived from the results for the information preservation metric as shown in Table 2. Here, we notice that a lot of handover information has actually been preserved. This highlights that the loss illustrated in Fig. 4 is mostly due to the substitution of low-granularity handover relations with handover relations that are on a higher level of generalization.

In contrast, the right graph in Fig. 4 illustrates the results obtained with PRIPEL using the weak configuration. Here, virtually all attribute values are connected. While this, trivially, preserves all existing handovers, it also introduces a large amount of false handovers. Arguably, this is a major loss of information. However, this result is expected for an anonymization technique that is based on noise insertion and that adopts randomization for the attribute values assigned to events.

Let us illustrate the differences between the two anonymization strategies with an exemplary analysis question. That is, Volvo IT, the company from which the BPIC 2013 log was obtained, was interested in understanding ping-pong behavior, i.e., cycles of handovers [28]. Approaching this question based on the attribute *org:role*, the original

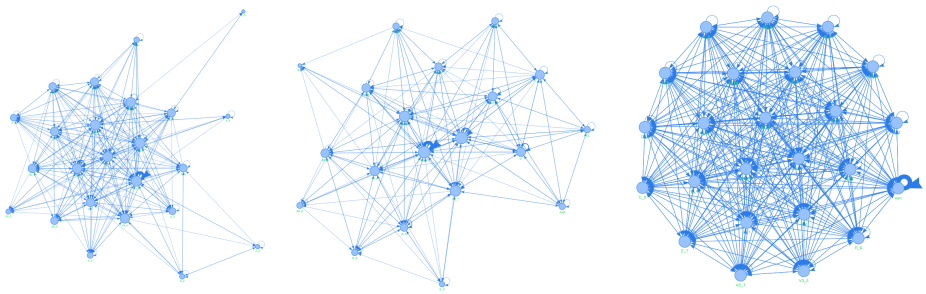


Fig. 4: (Left) Original Event Log, (Middle) $k = 5$, and (Right) PRIPEL weak setting handover graphs for the org:role attribute.

Table 2: Precision of generalized handovers

Log and Attribute	$k = 5$	$k = 10$	$k = 15$	$k = 20$
BPIC 2013 ”org:role”	85.2	80.0	79.3	79.4
BPIC 2013 ”organization involved”	100	100	100	100
BPIC 2013 ”resource country”	89.7	89.7	89.7	90.2
BPIC 2013 ”organization country”	89.2	88.1	88.1	87.6
CoSeLoG ”org:resource”	73.1	75.4	75.3	75.3

log reveals handovers between roles E10 and V3_2, while there are no connections for the pairs of roles {E9, V3_2}, {A2_1, C_1}, and {A2_2, C_1}. With PMDG, the roles E9 and E10 are generalized into a single role E*, which is connected to role V3_2. While this hides the fact that E9 was not connected to V3_2, it still suggests to assess the handovers of the set of E roles with V3_2. At the same time, the graph with PMDG does not include the incorrect handovers for A2 and C_1, so that these roles are not considered in the analysis of ping-pong behavior. The noisy result obtained with PRIPEL, in turn, is not suitable for this analysis, as it suggests that all roles are involved in cyclic handovers.

Process outcome prediction. Next, we consider the common task of process outcome prediction. Here, we look at the prediction of the ending activities using a decision tree classifier. In Fig. 5, we show the results from the classification experiment. The left heat maps show the results with different values for the privacy guarantee k. We observe that for both the BPIC 2013 log and the CoSeLoG log, higher privacy guarantees lead to better prediction metrics. This behavior is expected, since a more general log contains less control-flow variance, so that prediction becomes easier.

On the other hand, the classifiers trained based on event logs retrieved from PRIPEL provide classification results that have extremely low precision and recall. The results can be seen by observing the right heat maps. The noise inserted into the anonymized logs from PRIPEL clearly has a strong negative impact on the classification results and, hence, renders the anonymized event logs useless for outcome prediction.

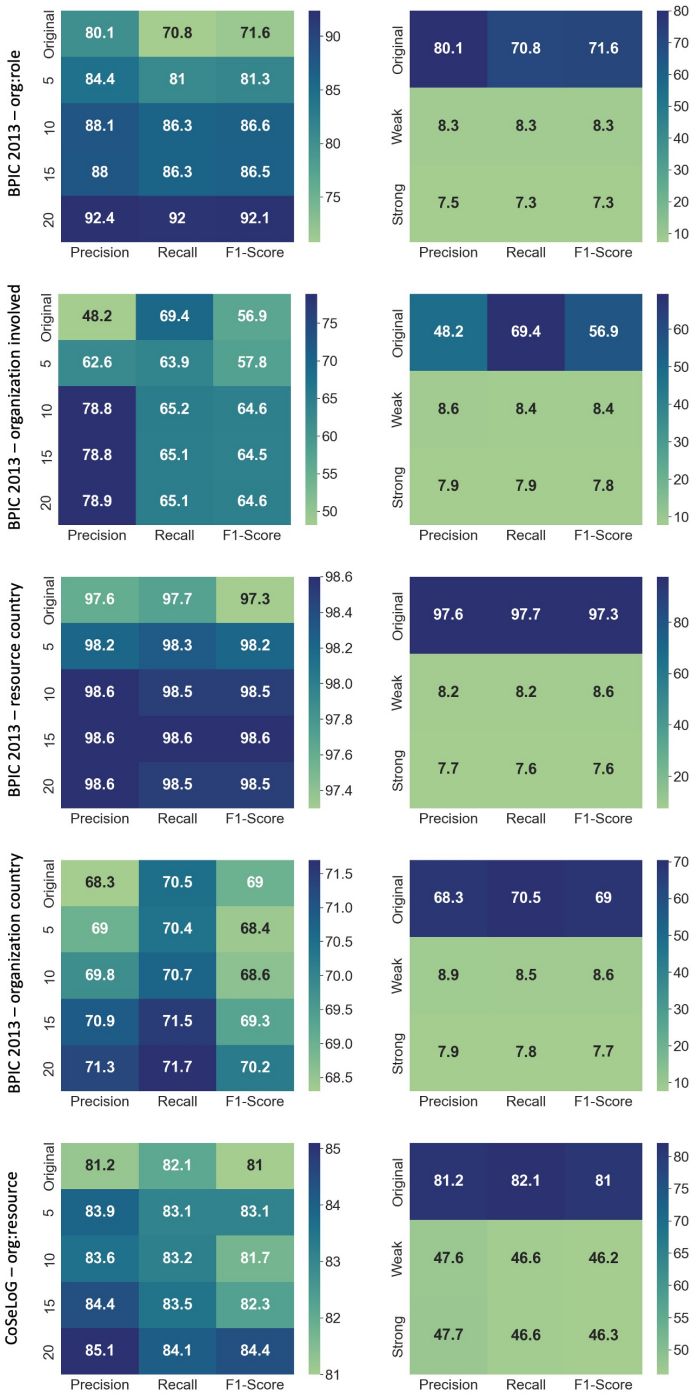


Fig. 5: Decision Tree Results for the **PMDG** (left) and **PRIPEL** (right) approaches.

6 Discussion

Below, we discuss several aspects of our approach, which will help to understand opportunities and limitations that need to be considered when applying PMDG.

Limitations of generalization. The main drawback of generalization is the loss of detail within the anonymized data. As a consequence, the data may no longer be useful for certain analyses. For instance, if individual resources are generalized towards their department, it is no longer possible to check whether the two-man rule was followed. As another example, consider the medical domain, where the dosage of a medication may be generalized. Questions related to the daily dosage limit may become impossible to answer under strong generalization.

The choice of generalization hierarchies. The success of applying PMDG highly depends on the generalization hierarchies that are available. Semantic hierarchies require manual work for their creation, a factor that can limit their availability. Also, for certain attributes, it might not be obvious how to generalize them. A prominent example are activities that often lack an unambiguous generalization hierarchy. Without the knowledge of a domain expert, it is not clear how to assess to which extent a generalization maintains process-specific information. Furthermore, PMDG makes no guarantee that the abstracted results will be useful in all situations. The usefulness of the results is dependent on the quality of the generalization hierarchy provided and the level of abstraction necessary to provide the privacy guarantee.

Different levels of abstraction. Based on the generalization technique used, an anonymized log might contain attribute values with differing levels of abstraction, i.e. an attribute encoding a region might contain values that represent a country or a continent. Mixing these different levels of abstraction can be challenging in the analysis, since most techniques do not offer built-in solutions to deal with such heterogeneous abstraction levels. Therefore, event logs that are anonymized with PMDG might require some post-processing before they can be utilized in common process mining solutions.

Risk of complete suppression. If an event log only consists of variants with a small number of traces that differ a lot in their attribute values, it is possible that these attribute values are essentially suppressed, i.e., generalizing a region from a value representing a city to the value *'World'*. In such a case, all potential benefits of generalization are lost. This problem can be addressed by providing hierarchies with a large number of generalization levels, so that the attribute values can converge to a level that still offers some utility. However, a large number of generalization levels may lead to an event logs with a lot of variance in its attribute values.

Curse of dimensionality. A well-known issue for achieving k-anonymity is the curse of dimensionality [32], meaning that an increase in attributes or events makes it harder to achieve the privacy guarantee. As we introduce additional attributes assigned to case and events, the data is partitioned into smaller equivalence classes. Consequently, an anonymized event log can be expected to lose more utility. A potential solution for this problem is the adoption of mixed privacy guarantees [33]. These techniques would allow for the use of noise-based anonymization for some attributes and generalization for others, while this choice is taken based on the requirements imposed in a specific analysis setting.

7 Conclusion

Within this work, we introduced PMDG, an anonymization framework that transforms events logs, so that they are protected by k-anonymity. The novelty our approach comes from (i) its ability to preserve the dependencies between different process perspectives as recorded in an event log, i.e. the control-flow and the attribute values assigned to events; and (ii) the the utilization of data generalization techniques as a means to achieve a privacy guarantee. In experiments with real-world event logs, we showed that PMDG outperforms the state of the art in terms of utility preservation for advanced process mining techniques. In future work, we intend to study how to support the construction and application of generalization hierarchies to optimize the utility of anonymized logs.

Acknowledgments

This work was supported by the German Federal Ministry of Education and Research (BMBF), grant number 16DII133 (Weizenbaum-Institute).

References

1. Gamal Elkoumy, Stephan A Fahrenkrog-Petersen, Mohammadreza Fani Sani, Agnes Koschmider, Felix Mannhardt, Saskia Nuñez Von Voigt, Majid Rafiei, and Leopold Von Waldthausen. Privacy and confidentiality in process mining: Threats and research challenges. *ACM TMIS*, 13(1):1–17, 2021.
2. Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.
3. Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, oct 2002.
4. Stephan A. Fahrenkrog-Petersen, Han van der Aa, and Matthias Weidlich. Pretsa: Event log sanitization for privacy-aware process discovery. *ICPM*, pages 1–8, 2019.
5. Majid Rafiei and Wil M.P. van der Aalst. Group-based privacy preservation techniques for process mining. *DKE*, 134:101908, 2021.
6. Felix Mannhardt, Agnes Koschmider, Nathalie Baracaldo, Matthias Weidlich, and Judith Michael. Privacy-preserving process mining. *BISE*, 61(5):595–614, 2019.
7. Adriano Augusto, Raffaele Conforti, Marlon Dumas, Marcello La Rosa, Fabrizio Maria Maggi, Andrea Marrella, Massimo Mecella, and Allar Soo. Automated discovery of process models from event logs: review and benchmark. *IEEE TKDE*, 31(4):686–705, 2018.
8. Weidong Zhao and Xudong Zhao. Process mining from the organizational perspective. In *Foundations of intelligent systems*, pages 701–708. Springer, 2014.
9. Irene Teinemaa, Marlon Dumas, Marcello La Rosa, and Fabrizio Maria Maggi. Outcome-oriented predictive process monitoring: Review and benchmark. *ACM TKDD*, 13(2):17:1–17:57, 2019.
10. Stephan A. Fahrenkrog-Petersen, Han van der Aa, and Matthias Weidlich. PRIPEL: privacy-preserving event log publishing including contextual information. In *BPM*, volume 12168 of *Lecture Notes in Computer Science*, pages 111–128. Springer, 2020.
11. Stephan Kessler, Jens Hoff, and Johann-Christoph Freytag. Sap hana goes private: from privacy research to privacy aware enterprise analytics. *VLDB*, 12(12):1998–2009, 2019.
12. Edgar Batista, Antoni Martínez-Ballesté, and Agusti Solanas. Privacy-preserving process mining: A microaggregation-based approach. *JISA*, 68:103235, 2022.
13. Stephan A Fahrenkog-Petersen, Martin Kabierski, Fabian Rösel, Han van der Aa, and Matthias Weidlich. Sacofa: Semantics-aware control-flow anonymization for process mining. In *ICPM*, pages 72–79. IEEE, 2021.

14. Gamal Elkoumy, Alisa Pankova, and Marlon Dumas. Mine me but don't single me out: Differentially private event logs for process mining. In *ICPM*, pages 80–87. IEEE, 2021.
15. Majid Rafiei and Wil Aalst. Mining roles from event logs while preserving privacy. 07 2019.
16. Edgar Batista and Agusti Solanas. A uniformization-based approach to preserve individuals' privacy during process mining analyses. *Peer-to-Peer Networking and Applications*, 14(3):1500–1519, 2021.
17. Majid Rafiei, Leopold von Waldthausen, and Wil MP van der Aalst. Supporting confidentiality in process mining using abstraction and encryption. In *SIMPDA*, pages 101–123. Springer, 2018.
18. Florence Corpet. Multiple sequence alignment with hierarchical clustering. *Nucleic acids research*, 16(22):10881–10890, 1988.
19. Ke Wang, Philip S Yu, and Sourav Chakraborty. Bottom-up generalization: A data mining solution to privacy protection. In *ICDM*, pages 249–256. IEEE, 2004.
20. Wai Kit Wong, Nikos Mamoulis, and David Wai Lok Cheung. Non-homogeneous generalization in privacy preserving data publishing. In *SIGMOD*, page 747–758, New York, NY, USA, 2010. ACM.
21. Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.
22. Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *SIGMOD*, pages 49–60, 2005.
23. Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE 2007, The Marmara Hotel, Istanbul, Turkey, April 15-20, 2007*, pages 106–115. IEEE Computer Society, 2007.
24. Fabian Prasser, Florian Kohlmayer, Ronald Lautenschläger, and Klaus A Kuhn. Arx-a comprehensive tool for anonymizing biomedical data. In *AMIA Annual Symposium Proceedings*, volume 2014, page 984. American Medical Informatics Association, 2014.
25. Anna Leontjeva, Raffaele Conforti, Chiara Di Francescomarino, Marlon Dumas, and Fabrizio Maria Maggi. Complex symbolic sequence encodings for predictive monitoring of business processes. In *BPM*, pages 297–313. Springer, 2016.
26. RP Jagadeesh Chandra Bose and Wil MP van der Aalst. Process diagnostics using trace alignment: opportunities, issues, and challenges. *Information Systems*, 37(2):117–141, 2012.
27. Thomas W Malone, Kevin Crowston, and George Arthur Herman. *Organizing business knowledge: The MIT process handbook*. MIT press, 2003.
28. Boudewijn F. van Dongen, Barbara Weber, Diogo R. Ferreira, and Jochen De Weerdt, editors. *Proceedings of the 3rd Business Process Intelligence Challenge co-located with 9th International Business Process Intelligence Workshop (BPI 2013), Beijing, China, August 26, 2013*, volume 1052 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.
29. M de Leoni and F Mannhardt. Road traffic fine management process, 2015. doi: 10.4121/uuid: 270fd440-1057-4fb9-89a9-b699b47990f5.
30. Joos Buijs. Receipt phase of an environmental permit application process ('WABO'), CoSeLoG project. *https://doi.org/10.4121/uuid:a07386a5-7be3-4367-9535-70bc9e77dbe6*, 2014.
31. Wil M. P. van der Aalst. *Process Mining: Data Science in Action*. Springer, Heidelberg, 2 edition, 2016.
32. Charu C Aggarwal. On k-anonymity and the curse of dimensionality. In *VLDB*, volume 5, pages 901–909, 2005.
33. Naoise Holohan, Spiros Antonatos, Stefano Braghin, and Pol Mac Aonghusa. (k, ϵ)-anonymity: k-anonymity with ϵ-differential privacy. *CoRR*, abs/1710.01615, 2017.