A Distance Measure for Privacy-preserving Process Mining based on Feature Learning

Fabian Rösel[1], Stephan A. Fahrenkog-Petersen[1],
Han van der Aa[2], and Matthias Weidlich[1]

[1] Humboldt-Universität zu Berlin , Berlin, Germany
[2] University of Mannheim, Mannheim, Germany
fabian.roesel@hu-berlin.de, stephan.fahrenkrog-petersen@hu-berlin.de,
han@informatik.uni-mannheim.de, matthias.weidlich@hu-berlin.de

Abstract. To enable process analysis based on an event log without compromising the privacy of individuals involved in process execution, a log may be anonymized. Such anonymization strives to transform a log so that it satisfies provable privacy guarantees, while largely maintaining its utility for process analysis. Existing techniques perform anonymization using simple, syntactic measures to identify suitable transformation operations. This way, the semantics of the activities referenced by the events in a trace are neglected, potentially leading to transformations in which events of unrelated activities are merged. To avoid this and incorporate the semantics of activities during anonymization, we propose to instead incorporate a distance measure based on feature learning. Specifically, we show how embeddings of events enable the definition of a distance measure for traces to guide event log anonymization. Our experiments with real-world data indicate that anonymization using this measure, compared to a syntactic one, yields logs that are closer to the original log in various dimensions and, hence, have higher utility for process analysis.

Keywords: Privacy · Anonymization · Trace Distance · Feature Learning

1 Introduction

Privacy-preserving process mining [9] aims at protecting personal data, while at the same time, enabling organizations to improve their business processes. The consideration of privacy in process mining is necessary, since the event logs used as a starting point for the analysis often include information about the individual people involved in process execution. For example, for a treatment process in a hospital, an event log may include personal information about the treated patients [26]. Since the respective data was commonly not recorded for the purpose of operational improvement, process mining is often considered a secondary use of the data and, as such, strictly regulated.

In general, there are two angles to approach privacy-preserving process mining [12]: One may anonymize the event log used for process mining [13] or one may design the algorithms for process mining such that the output satisfies privacy guarantees [14].

An established strategy for anonymizing event logs is to group traces together, i.e., to merge the data of multiple executions of a process. This way, the information about a single process instance and, hence, about an individual person involved therein, is not revealed. Such an approach is realized, for instance, in PRETSA [11] and the TLKC framework [24]. In order to preserve the utility of the event log for process analysis, these anonymization algorithms incorporate a distance measure. Intuitively, by merging traces that are close to each other, the resulting log will be close to the original one. Since process mining primarily targets the analysis of the behaviour exhibited by the traces, their closeness is captured in terms of their behavioural similarity.

Yet, existing algorithms [11,24] employ simple syntactic measures, such as the Levenshtein distance. This is a limitation, since, contrary to the assumption behind the Levenshtein distance, not all events have the same closeness. Arguably, the semantics of the activities referenced in the events in terms of their context suggests that some events are closer to each other than others. Neglecting such semantics means that any event log anonymization approach may induce a higher loss in utility than what would be necessary to achieve a certain privacy guarantee.

In this paper, we therefore investigate alternative approaches to measuring the distance of traces when anonymizing event logs. We study how embeddings of events, i.e., feature vectors that provide a semantic representation of the events, may be used as a foundation for a distance measure. Specifically, we rely on the Act2Vec [6] model to learn the embeddings from the original event log, thereby capturing the context of the activities of the events. Based thereon, we design a distance measure for traces that is tailored to event log anonymization, as it indicates which traces shall be merged into which other traces when aiming at a small loss of the log's utility for process analysis. Finally, we report on evaluation experiments to shed light on the impact of the adopted distance measure on the utility of an anonymized log. For several real-world event logs, we observe that using our semantic distance measure instead of a syntactic one, yields event logs that are closer to the original log along various dimensions.

In the remainder, we first discuss essential notions and notations (Section 2). We then present an embedding-based trace distance measure to use in event log anonymization (Section 3). Finally, we report on evaluation experiments (Section 4), before we review related work (Section 5) and conclude (Section 6).

2 Background

Below, we provide background for our work. In Section 2.1, we introduce a model for event logs. Section 2.2 outlines the PRETSA algorithm, a state-of-the-art algorithm for event log anonymization. In Section 2.3, we review the Act2Vec model to learn a feature-based representation of the events of a trace.

2.1 Event Logs

Process Mining relies on event logs that capture the execution of business processes [28]. We consider a common model of event logs, summarized as follows.

Table 1: Three example traces of a request handling process.

Case	Activity	Time	Case	Activity	Time	Case	Activity	Time
1	Check Loan Req.	9:05	2	Check Loan Req.	7:37	3	Check Loan Req.	9:49
1	Negotiate rate	10:04	2	Calculate rate	7:45	3	Report fraud	10:12
1	Set up contract	10:45	2	Set up contract	8:25	3	Block account	10:16
1	Inform client	14:08	2	Mail contract	9:50	3	Inform client	11:02

Each step of a process is represented by an activity $a \in \mathcal{A}$, with $\mathcal{A}$ denoting the universe of all activities. Each execution of such an activity is represented as an event $e = \langle c, a, ts \rangle$, where a is the respective activity, ts is a timestamp specifying the time of activity execution, and c is a case identifier signalling the instance of the process that the activity execution was related to. Moreover, by $\mathcal{E}$, we denote the universe of all events.

All events with the same case identifier form a trace, an ordered sequence of events $\langle e_1, \ldots, e_n \rangle = t \in \mathcal{E}*$, with $t(i) \in \mathcal{E}$, $1 \leq i \leq n$ denoting the i-th event of the trace. Here, the order of events in a trace is induced by their timestamps. We write $|t|$ for the trace length, i.e., the number of its events. A finite set of traces forms an event log, denoted by $L \subseteq \mathcal{E}*$.

Three traces of an exemplary event log for a process to handle loan requests are shown in Table 1. After a request has been checked, the interest rate is negotiated or calculated automatically, a contract is set up, and the client is informed or gets the contract posted by mail. However, the check may also identify fraud, so that the account is blocked, before the client is informed about it.

2.2 Event Log Anonymization with PRETSA

Striving for privacy-preserving process mining, an event log may be anonymized to avoid disclosure of sensitive information. This approach is realized in the *pre*fix-*t*ree-based event log *sa*nitization [11], or PRETSA for short. The algorithm transforms a given log to provide privacy guarantees based on k-anonymity [27] and t-closeness [16], while aiming to maximize the utility of the anonymized log.

The general idea behind PRETSA is protect the characteristics of a trace that enable the identification of an individual person. Specifically, it considers linkage attacks that are based on activity occurrences that directly point to individual persons, or exploit sensitive attributes for which differences in the observed distributions enable conclusions on individual persons. PRETSA guards against the former attack, by adopting a notion of k-anonymity, i.e., it forms groups of traces that are undistinguishable from each other and comprise at least k members. This way, the chances of linking a trace to an individual person is limited to $1/k$. To avoid linkage attacks through sensitive attributes, PRETSA also ensures t-closeness, i.e., it limits the difference in the distribution of a sensitive attribute of a group compared to the overall population.

To achieve the above guarantees, PRETSA transforms an event log by merging similar traces. It detects trace prefixes that violate the privacy guarantees and merges them into a similar variant. The latter is selected based on a closeness measure to limit the loss in utility caused by the transformation. However, PRETSA (and related approaches) assess the similarity of traces solely with a simple syntactic measure, i.e, the Levenshtein distance. Hence, all events of a trace are assumed to be equally close, which neglects the semantics of the respective activities, i.e., the context in which they are executed.

2.3 The Act2Vec Model

To incorporate the semantics of data elements, embeddings, i.e., feature vectors derived from a learned model, have been adopted in various domains. Most prominently, in natural language processing, an embedding can be constructed for each word of a text in order to represent its semantic meaning [19]. A model to derive the embedding of a word is learned by considering the surrounding words and, hence, captures the context in which a word typically appears.

The idea of word embeddings was recently lifted to event logs, as part of the so-called Act2Vec model [6]. This way, one may learn semantic representations of activities and, therefore, events. As in the case of word embeddings, these representations are learned from the context of an activity: If an event indicates the execution of a specific activity, the surrounding events in traces of the event log indicate which activities represent the common execution context. Consequently, closeness of activities (and hence, events) is assessed based on the closeness of their execution context, rather than other properties, such as the labels.

3 An Embedding-based Trace Distance Measure for Event Log Anonymization

This section introduces a distance function for traces that incorporates the semantics of activities and, hence, promises to induce a lower loss of utility when used in event log anonymization. First, we discuss the intuition of the measure (Section 3.1), before we turn to its definition (Section 3.2).

3.1 Intuition

Reconsider the three traces from Table 1. When comparing the first trace (Case 1) to the others, it is clear that Case 1 is more similar to the second trace than to the third: the first two both capture the regular handling of a loan request, differing solely in the way the interest rate is determined (negotiated versus automatically calculated) and in the method used to get back to the client (informed versus per mail). The third trace (Case 3), in turn, denotes a very different scenario, in which fraud was detected, resulting in the account being blocked.

In terms of the purely syntactic Levenshtein distance, however, the second and the third trace are equally distant from the first one. Both of them include two activity executions that are in line with the first trace, whereas each of them also includes two events that are without counterpart in the first trace. Therefore, existing anonymization techniques would consider both traces to be equally suitable in a merging step, even though merging the first and the third trace could be expected to lower the utility of the resulting log more drastically than the alternative solution of merging the first and the second trace.

Instead, when the semantics of activities in terms of their execution context are incorporated, a more suitable assessment of the traces' similarity may be achieved. For instance, the "*Negotiate rate*" and "*Calculate rate*" activities can be expected to have very similar execution contexts, i.e., after the request has been checked and directly preceding the activity to set up the contract. Hence, events referring to these activities are closer than events of less related activities, such as fraud reporting or account blocking.

3.2 A Distance Measure for Traces based on Embeddings

Feature learning based on the Act2Vec model [6] enables us to incorporate the semantics of the activities referenced in events in terms of their context of execution. The latter is induced by behavioural relations that describe common predecessors and successors of the respective activity. Since many process mining tasks are grounded in exactly these behavioural relations, preserving them as much as possible will increase a log's utility for process analysis.

Event distance. Since embeddings encode the context of activities referenced in events, i.e., their predecessors and successors, the similarity (or the distance, respectively) of two traces can be derived from the similarity (or distance) of their individual events. Therefore, we first define a distance for events, before using it as the basis to quantify the distance of traces.

We compare two events by the Cosine similarity of their feature vectors, which is defined over a vector space V as:

$$\text{cosS} : V \times V \to [-1, 1], \qquad (\mathbf{v}_1, \mathbf{v}_2) \mapsto \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{||\mathbf{v}_1||_2 \, ||\mathbf{v}_1||_2},$$

with $||\mathbf{v}||_2$ as the Euclidean norm of a vector $\mathbf{v}$. Note that two vectors $\mathbf{v}_1$, $\mathbf{v}_2$ are identical in direction, if $\text{cosS}(\mathbf{v}_1, \mathbf{v}_2) = 1$; orthogonal, if $\text{cosS}(\mathbf{v}_1, \mathbf{v}_2) = 0$, and opposing, if $\text{cosS}(\mathbf{v}_1, \mathbf{v}_2) = -1$.

In the remainder, given two events $e_1, e_2 \in \mathcal{E}$, we capture their distance based on embeddings as $d_e(e_1, e_2) = 1 - \text{cosS}(\text{act2vec}(e_1), \text{act2vec}(e_2))$.

On the direction of a trace distance for log anonymization. To assess the distance of two traces, the pair-wise distance of the events at the respective positions is calculated and summed up. Assuming that two traces t_1 and t_2 have equal length, this idea is illustrated in Figure 1. If two traces have a different length, the measure needs to take into account that, conceptually, events need to be added or removed to transform one trace into the other one. We therefore incorporate a penalty that is linear in the number of such excess events.

However, to use the distance measure to decide which traces to merge during event log anonymization, we define the penalty for excess events and, hence the trace distance measure, in an asymmetric manner. Algorithms for event log anonymization such as PRETSA merge one trace into another, which is

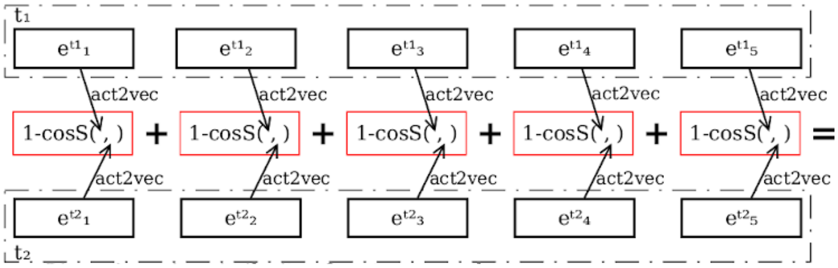

Fig. 1: Visual representation of the measure, assuming $|t_1| = |t_2|$.

an inherently directed operation. Here, it is desirable to avoid introducing new behaviour as part of the anonymization in order to maintain the log's utility for process analysis. Practically, PRETSA does not add new trace variants, i.e., all sequences of activity executions of traces in the anonymized log are already represented by at least one trace in the original log. As such, the algorithm guarantees that the successor relation over activities as derived from traces may only be reduced by the anonymization. The above is achieved by removing excess events of a trace when merging it into another one, which implies a potential loss of information on the context in which activities are executed. Hence, it is preferred to merge shorter traces into longer ones in order to maintain a larger part of the successor relation.

Moreover, the preservation of longer traces also helps to achieve privacy guarantees through the transformation. If traces of a certain variant, i.e., of a certain sequence of activity executions, are less than k-frequent, they have to be merged into other traces to achieve k-anonymity. Preferring the preservation of longer traces, there is a higher chance that these traces can be merged into traces that include the respective sequence of activity executions already as a prefix. In that case, no information in terms of the successor relation would be lost.

Trace distance. Following the above arguments, we propose a directed distance measure for traces to guide event log anonymization. Given two traces t_1 and t_2, this distance incorporates that, if both traces have different lengths, merging the shorter into the longer one is preferred. The distance $d(t_1, t_2)$ is defined as:

$$d(t_1,t_2) = \begin{cases} \sum\limits_{1 \leq i \leq |t_1|} d_e(t_1(i), t_2(i)) & \text{if } |t_1| = |t_2| \\ \sum\limits_{1 \leq i \leq |t_2|} (d_e(t_1(i), t_2(i))) + (|t_1| - |t_2|) \cdot \rho_A & \text{if } |t_1| > |t_2| \\ \sum\limits_{1 \leq i \leq |t_1|} (d_e(t_1(i), t_2(i))) + (|t_2| - |t_1|) \cdot \rho_R & \text{if } |t_1| < |t_2| \end{cases}$$

where $\rho_A, \rho_R \in \mathbb{N}$ are penalties for event addition and event removal, respectively, such that $\rho_A > 1$ and $\rho_R > \rho_A$.

As the distance measure incorporates the direction in which traces are potentially merged, the penalty ρ_R for event removal shall be larger than the one for event addition, ρ_A. Empirically, we found that a 50% increase of the penalty yielded the best results. Moreover, both penalties shall dominate the distance of any pair of individual events. Hence, we set $\rho_A = 2$ and $\rho_R = 3$ as default values.

4 Evaluation

In the following section, we present an evaluation of our approach. The aim of this evaluation is to answer the following research questions:

RQ1: Can we preserve more control-flow information by incorporating an embedding-based trace distance measure into event log anonymization?
RQ2: Do we preserve less utility in aspects unrelated to control-flow by optimizing for higher control-flow related utility?
RQ3: How do the internal properties of the distance measure impact the preserved utility?

To answer these questions, we apply our approach on the logs presented in Section 4.1, before Section 4.2 lays out the experimental setup. Section 4.3 presents and interprets the obtained results.

4.1 Dataset

We perform the evaluation on three real-world event logs. As shown in Table 2, the logs differ considerably in their domains and structuredness. The *CoSeLoG* log captures a relatively structured process, concerning the application of environmental permits, whereas the *Sepsis* log covers an unstructured process corresponding to clinical pathways from a hospital. Finally, *BPIC 2020* log captures the semi-structured reimbursement process of travel costs at a Dutch university. While all three event logs have comparable size in terms of the number of cases, and hence traces, the differences between them in terms of structuredness are clearly illustrated by the average and maximum numbers of cases per trace variant. In *CoSeLoG*, hundreds of traces may follow the same sequence of activity executions, whereas in *Sepsis*, traces often show a unique activity sequencing.

Table 2: Event log characteristics.

Event log	**Cases**	**Variants**	**avg. cases/var.**	**max. cases/var.**
CoSeLoG [2]	1,434	116	12.4	713
Sepsis [3]	1,050	846	1.2	35
BPIC 2020 [1]	2,099	896	2.3	206

4.2 Experimental Setup

Configuration. In our experiments, we compare the proposed method of incorporating the embedding-based distance measure against the original PRETSA using Levenshtein distance. To vary the required privacy guarantees, k-anonymity and t-closeness, we evaluate all combinations with $k = 2^i$ for $i \in [1, 8]$ and $t \in \{0.1, 0.25, 0.5, 0.75, 1.0\}$, yielding a total of 40 different settings per event log.

Evaluation measures. To asses the utility of anonymized event logs, we use two metrics to analyse the preservation of control-flow behaviour and one metric to measure the impact on the average cycle time of each activity:

Advanced Behavioural Appropriateness. To measure the impact that anonymization has on the control-flow, we first use advanced behavioural appropriateness [25], which encodes behaviour within a log according to binary activity relations. Specifically, given a log, an activity A may $\{$*always, sometimes, never*$\}$ indirectly $\{$*follow, precede*$\}$ another activity B. By comparing these relations in an anonymized event log and its original counterpart, the metric yields a similarity score $\in [0,1]$, where 1 represents total equality for every relation between activity pairs in the two event logs and 0 if none of the relations are equal.

Truly sampled Behaviour. We also use *truly sampled behaviour* [15] as a metric to measure the preservation of control-flow information. While the behavioural appropriateness considers indirectly-follows relations between activities, this measures looks at directly-follows relations and also consider their relative frequency. Specifically, it quantifies what fraction of directly-follows relations of the original log are *appropriately sampled*. It considers a directly-follows relation appropriately (or truly) sampled when, adjusted for potentially different log size, it appears approximately the same time in both logs. The *Truly Sampled Score* then is the proportion of truly sampled directly-follows relations. Therefore a Truly Sampled Score of 100% is considered optimal, while a score of 0% indicates no relation-wise similarity between the compared event logs.

Total Duration Error. Finally, we also investigate how our approach impacts the duration of an activity, which can be regarded as a sensitive attribute that shall be anonymized in accordance with a specified t-closeness guarantee. For this, we compute the total duration error. It is a simple metric based on the assumption that similar activities take a similar amount of time to execute. The error is calculated by comparing the total time it takes to execute all traces in the sanitized log to the total execution time of the original log.

Implementation. To conduct our experiments, we implemented PRETSA using the embedding-based distance measure, as well as all used evaluation measures in Python. The source code is available at GitHub[3] under MIT license.

4.3 Results

To answer RQ1, we consider the evaluation metrics related to the control-flow behaviour of the log. Figure 2 illustrates that our feature-based distance measure (FDM) can improve Advanced Behavioural Appropriateness scores for certain anonymization parameters in two of the three real-life event logs. Best results were achieved using the *Sepsis* log, while *CoSeLoG* yielded partially better results. *BPIC 2020* performed approximately equal for both distance measures.

We see similar results when considering the Truly Sampled Score. Again, *Sepsis* shows the biggest improvements, for a range of k and t, whereas *CoSeLoG* improved for selected parameter combinations. No noticeable differences can be observed for *BPIC 2020*. Hence, regarding RQ1, we can say, that our embedding-based measure can preserve more control-flow utility. However, applying it gives no guarantee of achieving this aim.

[3] https://github.com/roeselfa/FeatureLearningBasedDistanceMetrics

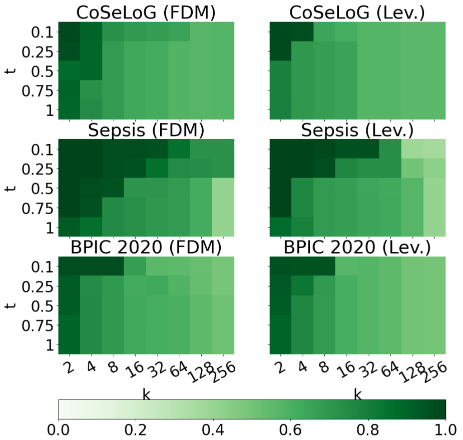

Fig. 2: Advanced Behavioural Appropriateness of anonymized logs.

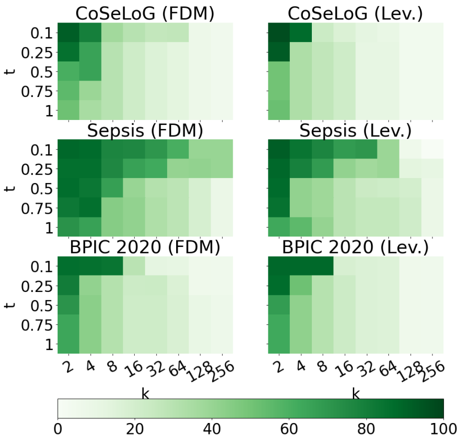

Fig. 3: Truly Sampled scores of anonymized logs.

Additionally, we want to investigate, if the application of the embedding-based measure harms other utility aspects of the log (RQ2). By considering the duration errors depicted in Figure 4, we observe the following: For *Sepsis*, while results for specific scenarios vary widely, showing huge improvements in some cases and comparable declines in others, the global error neither increased, nor decreased. Rather negligible improvements were made for *CoSeLoG*, whereas both approaches are comparable for *BPIC 2020*. Concluding, in regards to RQ2, our results do not provide evidence that applying a distance measure focused on control-flow aspects will have negative implications for other utility aspects of the event log.

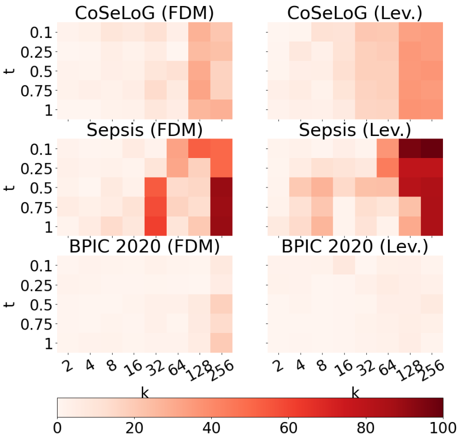

Fig. 4: Total Duration Error of anonymized logs.

Finally, to address RQ3, we take a closer look at the internal properties of our distance measure. The embedding-based distance measure captures the context of each event in a feature vector. Similar feature vectors of two events indicate that the activities referenced in the events often appear in a similar context and thus, are semantically exchangeable to a certain extent. An event log with many similar events therefore enables the embedding-based distance measure to guide anonymization effectively in the selection of traces to merge, whereas event logs where events relate to activities of disjunct contexts, lack such possibility.

To investigate this, we first consider the average and standard deviations of the computed event distances per log, shown in Table 3. The table reveals that events in *BPIC 2020* are particularly distant from each other, with the highest average (0.93) and smallest standard deviation (0.28), whereas the *Sepsis* log has the lowest average and highest standard deviation. This reveals that for the latter, our approach using the embedding-based distance is able to have a greater impact on the anonymization procedure when compared to an approach based on the Levenshtein distance, which is indeed confirmed by the larger positive impact that our approach has for this log, when compared to the *BPIC 2020* and *CoSeLoG* logs.

Table 3 furthermore shows the median share of the x most frequent directly followers per activity (as a fraction of its total number of directly followers). This way, we measure how similar the contexts of activities are in a log. The activities of *BPIC 2020* have the most common context, backing up the high distinctiveness of them. The *Sepsis* log on the other hand, are on average noticeably closer in distance and share less neighbouring events. Allowing the measure to distinguish stronger between the closeness of activities. Consequently, we saw higher utility preservation in the previous experiments for the *Sepsis* log, compared to the Levensthein distance-based baseline. With *CoSeLoG* falling both in terms of utility preservation and internal measures between the two other logs.

Table 3: Internal properties of our measure in terms of the distance between events and the share of the top-x most frequent followers per activity (median)

	Event distance		**Share of top-x followers**		
Event log	**Avg.**	**Stdev.**	**x = 1**	**x = 2**	**x = 3**
CoSeLoG	0.83	0.29	82.8%	91.3%	99.2%
Sepsis	0.77	0.42	67.0%	82.1%	94.3%
BPIC 2020	0.93	0.28	88.8%	96.7%	99.0%

5 Related Work

Privacy-preserving process mining received much attention recently [9]. In particular, the problem of anonymizing event logs was addressed in several approaches, since logs generally show serious re-identification risks [29]. PRETSA [11] has been proposed as an algorithm to tackle this problem based on the privacy notions of k-anonymity and t-closeness. Similarly, Rafiei et al. [24,23] proposed alternative techniques to ensure a group-based privacy guarantee, inspired by k-anonymity. A similar approach to group-based privacy protection is also realized in [4]. Either way, event log anonymization relies on a distance measure to decide on which traces to merge with each other and, so far, only the Levenshtein distance as a simple syntactic measure was considered. We addressed this shortcoming with the embedding-based measure presented in this work.

Several studies also focused on differential privacy, an alternative privacy guarantee, that limits the impact one individual may have on the data. While [18] and [14] focus on specific queries performed on the event logs, the issue of publishing differential private event logs was covered by [13] and [10].

Instead of anonymizing event logs, privacy-preserving algorithms for process mining may be employed to protect the data of individual persons [12]. Here, the distributed analysis of the control-flow of event logs was explored in [7] and [17], whereas privacy-aware role mining was studied in [21].

The application of privacy-preserving process mining in the healthcare domain was studied in [20], highlighting the importance of the ability to customize the algorithms to domain-specific requirements. To foster the uptake of privacy-preserving process mining, the respective techniques have been made accessible as tools, including ELPaaS [5], Shareprom [8], and the tool described in [22].

6 Conclusion

In this paper, we presented an embedding-based trace distance measure that is tailored to event log anonymization. It leverages feature vectors learned by the Act2Vec model to assess the similarity of events and, hence, traces. Unlike syntactic distances commonly used in event log anonymization, it thereby incorporates the context in which events occur. Moreover, the measure considers differences in the trace lengths in an asymmetric manner to guide the selection of which trace to merge into which other trace as part of the anonymization. Our experimental results indicate that an embedding-based distance measure can indeed improve the results of event log anonymization, compared to the use of the Levenshtein distance. Specifically, the improvement is most pronounced for event logs that contain events of different activities that frequently appear in similar contexts.

References

1. BPI challenge 2020: Prepaid travel costs. https://data.4tu.nl/articles/dataset/BPI_Challenge_2020_Prepaid_Travel_Costs/12696722, accessed: 2020-05-12.
2. Receipt phase of an environmental permit application process (‘wabo’), coselog project. https://data.4tu.nl/collections/Environmental_permit_application_process_WABO_CoSeLoG_project/5065529, accessed: 2020-05-11.
3. Sepsis cases - event log. https://data.4tu.nl/articles/dataset/Sepsis_Cases_-_Event_Log/12707639, accessed: 2020-04-03.
4. Batista, E., Solanas, A.: A uniformization-based approach to preserve individuals' privacy during process mining analyses. Peer Peer Netw Appl pp. 1–20 (2021)
5. Bauer, M., Fahrenkrog-Petersen, S.A., Koschmider, A., Mannhardt, F., van der Aa, H., Weidlich, M.: ELPaaS: Event log privacy as a service. In: BPM Demos. pp. 159–163 (2019)
6. De Koninck, P., vanden Broucke, S., De Weerdt, J.: act2vec, trace2vec, log2vec, and model2vec: Representation learning for business processes. In: BPM. pp. 305–321. Springer (2018)
7. Elkoumy, G., Fahrenkrog-Petersen, S.A., Dumas, M., Laud, P., Pankova, A., Weidlich, M.: Secure multi-party computation for inter-organizational process mining. In: BPMDS, pp. 166–181. Springer (2020)

8. Elkoumy, G., Fahrenkrog-Petersen, S.A., Dumas, M., Laud, P., Pankova, A., Weidlich, M.: Shareprom: A tool for privacy-preserving inter-organizational process mining. In: BPM Demos. pp. 72–76 (2020)
9. Elkoumy, G., Fahrenkrog-Petersen, S.A., Sani, M.F., Koschmider, A., Mannhardt, F., von Voigt, S.N., Rafiei, M., von Waldthausen, L.: Privacy and confidentiality in process mining–threats and research challenges. arXiv:2106.00388 (2021)
10. Elkoumy, G., Pankova, A., Dumas, M.: Mine me but don't single me out: Differentially private event logs for process mining. arXiv:2103.11739 (2021)
11. Fahrenkrog-Petersen, S., van der Aa, H., Weidlich, M.: PRETSA: Event log sanitization for privacy-aware process discovery. In: ICPM (2019)
12. Fahrenkrog-Petersen, S.A.: Providing privacy guarantees in process mining. In: CAiSE (Doctoral Consortium). pp. 23–30 (2019)
13. Fahrenkrog-Petersen, S.A., van der Aa, H., Weidlich, M.: PRIPEL: privacy-preserving event log publishing including contextual information. In: BPM. pp. 111–128 (2020)
14. Kabierski, M., Fahrenkrog-Petersen, S.A., Weidlich, M.: Privacy-aware process performance indicators: Framework and release mechanisms. In: CAiSE (2021)
15. Knols, B., van der Werf, J.M.E.M.: Measuring the behavioral quality of log sampling. In: ICPM. pp. 97–104 (2019)
16. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: ICDE. IEEE (2007)
17. Liu, C., Duan, H., Zeng, Q., Zhou, M., Lu, F., Cheng, J.: Towards comprehensive support for privacy preservation cross-organization business process mining. IEEE Transactions on Services Computing **12**(4), 639–653 (2016)
18. Mannhardt, F., Koschmider, A., Baracaldo, N., Weidlich, M., Michael, J.: Privacy-preserving process mining. BISE **61**(5), 595–614 (2019)
19. Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: NAACL. pp. 746–751 (2013)
20. Pika, A., Wynn, M.T., Budiono, S., Ter Hofstede, A.H., van der Aalst, W., Reijers, H.A.: Privacy-preserving process mining in healthcare. International journal of environmental research and public health **17**(5), 1612 (2020)
21. Rafiei, M., van der Aalst, W.: Mining roles from event logs while preserving privacy. In: BPM Workshops. pp. 676–689 (2019)
22. Rafiei, M., van der Aalst, W.: Practical aspect of privacy-preserving data publishing in process mining. In: BPM Demos. pp. 92–96 (2020)
23. Rafiei, M., van der Aalst, W.: Group-based privacy preservation techniques for process mining. arXiv preprint arXiv:2105.11983 (2021)
24. Rafiei, M., Wagner, M., van der Aalst, W.: TLKC-privacy model for process mining. In: RCIS. pp. 398–416. Springer (2020)
25. Rozinat, A., Aalst, W.: Conformance checking of processes based on monitoring real behavior. Information Systems **33**, 64–95 (03 2008)
26. Stefanini, A., Aloini, D., Benevento, E., Dulmin, R., Mininno, V.: Performance analysis in emergency departments: a data-driven approach. Measuring Business Excellence (2018)
27. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10, no. 05**, 557–570 (2002)
28. Van Der Aalst, W.: Process mining: Overview and opportunities. ACM Transactions on Management Information Systems (TMIS) **3**(2), 1–17 (2012)
29. von Voigt, S.N., Fahrenkrog-Petersen, S.A., Janssen, D., Koschmider, A., Tschorsch, F., Mannhardt, F., Landsiedel, O., Weidlich, M.: Quantifying the re-identification risk of event logs for process mining. In: CAiSE. pp. 252–267. Springer (2020)