Differentially Private Event Logs with Case Attributes

Hannes Ueck[1], Robert Andrews[2], Moe T. Wynn[2], and Sander J. J. Leemans[1,3]

[1] RWTH Aachen, Germany
[2] Queensland University of Technology, Australia
[3] Fraunhofer FIT, Germany
hannes.ueck@rwth-aachen.de, s.leemans@bpm.rwth-aachen.de

Abstract. Event logs capture the execution of processes, record activities and additional information. A trace represents a single instance of a process and includes a sequence of activity records and case attributes with additional information. Event logs may contain sensitive personal information that could harm an individual's privacy if it is published without pre-processing. Differential privacy (DP) limits the disclosure of new information about any individual when publishing an event log beyond the publicly available background knowledge. Many privacy-preserving approaches to event log publishing ensure DP. Traditional methods focus on preserving the control flow but omit case attributes, limiting comprehensive process analysis based on these attributes. This work addresses this limitation by proposing a novel privacy-preserving event log publishing framework. Our approach ensures privacy for the control flow and case attributes, utilising synthetic tabular data generation approaches based on machine learning that guarantee DP. The framework allows for the use of various tabular data generation approaches. Experimental results with real-world event data demonstrate the framework's feasibility and highlight the trade-off between data utility and the guaranteed levels of privacy.

Keywords: Differential Privacy · Process Mining · Event Logs · Machine Learning.

1 Introduction

Modern information systems are designed to record the activities executed in processes across different domains such as businesses or healthcare [1]. This information is consolidated into event logs, which are collections of the executed activities. Each activity is part of a sequence of activities that comprise a case and additional attributes on the case level might be recorded.

Process mining aims to generate insights from event logs [1]. This could include discovering a process's control flow, optimising it, detecting deviations from the expected behaviour, or predicting future activities. Combined with case attributes more detailed analyses are possible, e.g. predictive monitoring of processes [19] or identifying differing process behaviour for cohorts of patients [15].

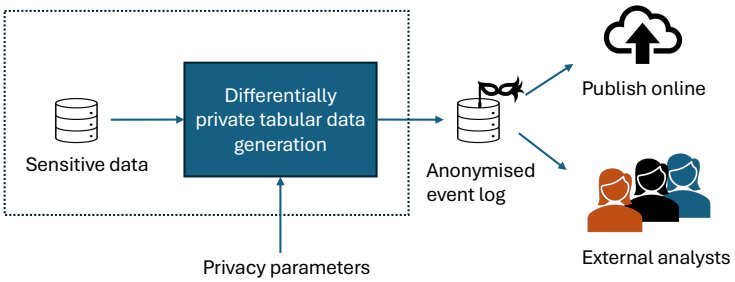

Fig. 1: Conceptual overview

Event logs contain personal data about the individuals involved in the process. Legislation enforces the protection of personal data, the General Data Protection Regulation (GDPR) in the European Union (*Regulation 2016/679, European Parliament*) or the Privacy Act in Australia (*Privacy act 1988, Commonwealth of Australia*). Personal data is considered sensitive and thus needs additional protection when it reveals, for instance, health-related information, ethnic origin or political opinions of individuals. Event logs contain possibly sensitive information in the recorded activities or information included in the additionally collected attributes. For a hospital process, this might be a patient's treatment, the time when the treatment is executed, or the patient's treatment outcome. An adversary might be able to link an individual to a case and re-identify the individual. To protect the privacy of the individuals, it is necessary to apply transformations to the data before publishing it [6,17].

After applying the transformations, the data should still be useful for subsequent analysis [12]. It has been shown that simple removal of names and unique identifiers from the data might lead to re-identification [25]. Differential privacy (DP) provides mathematically proven privacy that limits the impact of any single individual on a dataset [3]. Consequently, publishing a differentially private dataset offers only negligible additional information to adversaries beyond general background knowledge. Unlike group-based techniques where records are generalised or suppressed to create groups of similar records [26], DP also prevents predicate singling out attacks, to which group-based techniques are vulnerable [2].

Most existing methods to guarantee DP for event logs focus on the control flow by omitting other information contained in the event log [8,9,17]. [10] proposed a method to include contextual information, timestamps and case attributes in the anonymised event log. However, this method assumes that the case attributes are independent, which is unrealistic for most common event logs. To the best of our knowledge, no other approaches support the privatisation of an event log with case attributes while guaranteeing DP.

Methods to guarantee DP for purely tabular data have been the focus of research [18,27,31]. These tabular data generation algorithms (TDG) reproduce a dataset by estimating the underlying distribution of the data [31]. Since this

does not inherently ensure DP, noise is added to the estimation process to achieve it. We argue that because of the similarities between tabular data and event log data, applying those proven approaches to event log data is possible. However, we argue that additional measures are required to ensure that the characteristics of event logs are accounted for.

In this paper, we propose a framework to apply synthetic tabular data generation methods to event logs while guaranteeing DP for the generated event log. Our contribution is twofold: (1) We propose a framework (DP-ELCA) to anonymise event logs with case attributes while guaranteeing DP. (2) We discuss the privacy implications when applying tabular data generation techniques to event logs. (3) We benchmark our framework using different TDGs on multiple real-life event logs and assess the similarity of the anonymised event log to the original data.

The remainder of this paper is structured as follows. Section 2 introduces the necessary background information. In Section 3 we present our approach. We evaluate our approach in Section 4. In Section 5 we discuss related work and conclude in Section 6.

2 Preliminaries

2.1 Event logs

An event log L is composed of traces, where each trace consists of a sequence of activities. The traces represent the executions of cases in the process and next to the sequence of activities each trace can have case attributes that provide additional information about the case. We consider only the control flow information and the case attributes in this approach.

Definition 1 (Event log). *An event log L with case attributes is defined as a set of tuples, where each tuple consists of a trace and a set of case attributes: $L = \{(\sigma_1, \mathbf{ca_1}), (\sigma_2, \mathbf{ca_2}), \ldots, (\sigma_n, \mathbf{ca_n})\}$. Each trace σ is a sequence of activities $\sigma_i = \langle a_1, a_2, \ldots, a_l \rangle$, where a_j is the j-th activity in the trace and l its varying length. The set of case attributes $\mathbf{ca_i} = \{att_i^1, att_i^2, \ldots, att_i^p\}$, where p is the number of case attributes, provides additional information about the case.*

2.2 Differential privacy

Differential privacy (DP) is a probabilistic privacy guarantee given on the output of a data processing mechanism $\mathcal{M}$ given an input in form of a dataset $\mathbf{X}$ [3]. It states that anyone examining the output draws the same conclusions about an individual's private information regardless of whether that individual's data was part of the input to the mechanism or not. We apply the definition of DP in [3] to event logs.

Definition 2 (ε, δ-DP for event logs). *Let L_1 and L_2 be two event logs differing in at most one trace σ. Further, let $\varepsilon > 0$ and $\delta \in [0, 1]$ be privacy parameters.*

Then a randomised mechanism $\mathcal{M}$ provides (ε, δ)-DP if for all subsets S of the output space of $\mathcal{M}$,

$$\frac{\Pr[\mathcal{M}(L_1) \in S]}{\Pr[\mathcal{M}(L_2) \in S]} \leq e^{\varepsilon} + \delta \tag{1}$$

where the probability Pr *is taken over the randomness introduced by the mechanism $\mathcal{M}$.*

Intuitively, ε represents the privacy loss incurred when including an individual's data in the dataset. δ is the probability for a deviation from this guarantee.

Proven theorems give bounds for the privacy budget when applying multiple differentially private mechanisms to the same dataset. The k-fold adaptive composition theorem allows chained heterogeneous mechanisms to access the outputs of the previous mechanisms:

Theorem 1 (k-fold adaptive composition [13]). *Let $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ be $(\varepsilon_i, \delta_i)$-differentially private mechanism for $i \in [k]$, $\varepsilon_i > 0$, $\delta_i \in [0, 1]$ and $\tilde{\delta} \in [0, 1]$. Then the combined mechanism using k-fold adaptive composition of $\mathcal{M}_i$ provides $(\tilde{\varepsilon}_{\tilde{\delta}}, 1 - (1 - \tilde{\delta}) \prod_{i=1}^{k} (1 - \delta_i))$-DP with $\tilde{\varepsilon}_{\tilde{\delta}} =$*

$$\begin{aligned} \min \Bigg\{ & \sum_{i=1}^{k} \varepsilon_i, \sum_{i=1}^{k} \frac{(e^{\varepsilon_i} - 1)\varepsilon_i}{e^{\varepsilon_i} + 1} + \sqrt{\sum_{i=1}^{k} 2\varepsilon_i^2 \log\left(e + \frac{\sqrt{\sum_{i=1}^{k} \varepsilon_i^2}}{\tilde{\delta}}\right)}, \\ & \sum_{i=1}^{k} \frac{(e^{\varepsilon_i} - 1)\varepsilon_i}{e^{\varepsilon_i} + 1} + \sqrt{\sum_{i=1}^{k} 2\varepsilon_i^2 \log\left(\frac{1}{\tilde{\delta}}\right)} \Bigg\} \end{aligned} \tag{2}$$

2.3 TraVaS: Differentially private trace variant selection

TraVaS is a framework that allows for releasing the distribution of trace variants based on a private partition selection algorithm with a privacy budget of (ε, δ). It utilises a k-Truncated Symmetric Geometric Distribution (k-TSGD) to add noise to the frequencies of the trace variants. Based on the privacy budget noise values are drawn from the k-TSGD and added to each trace variant count. The output then only contains trace variants where the perturbed count is above a threshold that is calculated using the privacy budget. This ensures DP for the resulting trace variant distribution [23].

2.4 Tabular data generation

Tabular data generation aims to generate synthetic data that closely resembles the statistical properties of the original data [30]. It can be used to provide data for analysis without revealing sensitive information about the individuals in the dataset. While the original data is not disclosed, the synthetic data should keep the statistical properties of the original data, such as the distribution of the data points and the correlations between the columns.

Methods based on Graphical Models The methods based on graphical models first measure the conditional distributions in the dataset [18,31]. Noise

is introduced into the measurements to achieve DP, where the scale of the noise is computed based on the provided privacy parameters. Then, the parameters of a probabilistic graphical model are estimated based on the noisy measures. Finally, the estimated model is sampled to generate an anonymised dataset. This ensures the privacy of individuals in the dataset while allowing the synthetic data to retain the properties of the original data.

Methods based on generative neural networks Generative neural networks are used to generate data that resembles the features of the original data [29]. Differentially private stochastic gradient descent (DP-SGD) is used to train the models to achieve DP. This includes adding noise to the gradients and clipping them to ensure that the model does not learn exact copies of data points in the original dataset.

3 Approach: DP-ELCA

DP-ELCA, as shown in Figure 2, guarantees (ϵ, δ)-DP for the control flow and the case attributes, while ensuring the utility of the anonymised event log by keeping the statistical properties of the original event log.

3.1 Transforming the Event Log to Tabular Data

The event log is transformed into a tabular data format by aggregating the event log on the case level. Each case in the event log corresponds to a row in the tabular data and each row contains a list of activities, i.e. the variant, and the case attributes.

Definition 3 (Aggregated event log). *Let L be an event log and $\mathbf{X}$ the tabular dataset constructed from L. For each tuple $(\sigma_i, \mathbf{ca_i}) \in L$, a datapoint $\mathbf{x_i} \in \mathbf{X}$ is created. A datapoint is the concatenation of the numerical representation of the trace variant $c_i^v = f(\sigma_i)$ with k case attributes in $\mathbf{ca_i}$. Thus, $\mathbf{x_i} = c_i^v \circ c_i^{att^1} \circ c_i^{att^2} \circ \cdots \circ c_i^{att^k}$.*

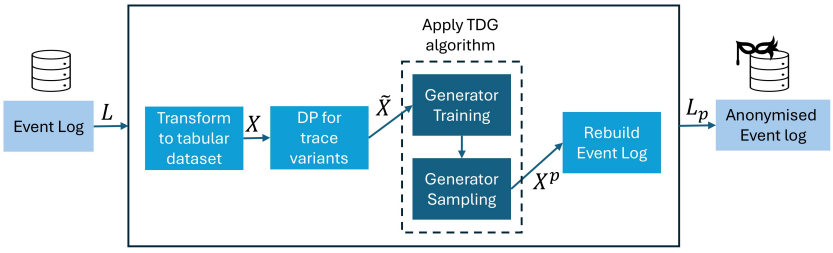

Fig. 2: Overview of the DP-ELCA.

3.2 Privacy implications of trace variants for private tabular data generation

In traditional tabular data generation settings, it is assumed that the domains of the columns of the dataset are public knowledge, meaning that for a dataset $\mathbf{X}$ with l columns and n rows, $dom(c_j)$ for $j \in [l]$ are publicly known. The algorithms are applied to datasets where the usual range of values or categories within the columns are known and do not depend on the dataset, e.g., the range of blood pressure measurements or categories of diseases. For a categorical column $dom(c_c)$, it is assumed that there are many more rows of data than the number of categories ($|dom(c_c)| << n$).

Given these assumptions, differentially private TDG algorithms reproduce the exact domains of the original dataset in the generated data ($dom(c_j) = dom(c_j^p), j \in [l]$). For datasets that are derived from event logs as defined in Definition 3, this assumption of independence holds for the columns $c^{att^k}, i \in [k]$ that contain data taken directly from the case attributes. However, for c_v, which contains the categories of the trace variants, this assumption does not hold. The domain of the trace variants $dom(c_v)$ is dependent on the traces contained in the event log.

Consider a scenario where an attacker knows the trace variant related to an individual and knows that only this individual could produce this trace variant. Then there are two ways how the attacker could infer information:

(1) If the individual is in the event log, the anonymised event log would contain cases with such a trace variant. In that case, the attacker could infer that the individual is included in the event log.
(2) If the individual is not in the event log, the anonymised event log would not include cases with such a trace variant. In that case, the attacker could infer that the individual was not part of the event log.

This scenario illustrates that information about an individual could be leaked if the algorithm does not change the domain of the attributes in the output ($dom(c_v) = dom(c_v^p)$). The same problem holds for infrequent trace variants where the existence of a group of individuals in the dataset could be leaked. Therefore, filtering infrequent trace variants is needed to ensure privacy. Note that the information leakage is independent of the privacy budget (ϵ, δ) because it has no influence on the domain produced by the TDG. We showed that after filtering out infrequent traces the attacker cannot single out an individual or a group of individuals as described above.

We conclude that directly applying traditional tabular data generation approaches is possible but needs a filtering step beforehand. For this reason our approach applies a DP trace selection algorithm [23] before applying the TDG.

3.3 DP-ELCA

The framework $\mathcal{F}$ receives an event log L as input and generates a differentially private event log L_p as output. Figure 2 shows the four steps of the framework. First, for each case, the sequence of activities and the case attributes

are extracted from L. Next, infrequent sequences of activities are filtered out, using [23], and a table is created. Second, a TDG is trained to estimate the statistical properties of the tabular data. Third, synthetic tabular data is sampled from the estimated model. The generated tabular data is then transformed into an event log containing the sequences of activities and case attributes. Next, we introduce each of the steps in more detail.

Transform event log to tabular data. This step involves the transformation of the event log L to a tabular dataset $\mathbf{X}$. According to Definition 3, the tabular dataset is constructed by aggregating the event log by case identifier. We build a look-up table to convert the trace variants into their numerical representation. Further, we let the user choose which case attributes to include in the tabular dataset $\mathbf{X}$. Any directly identifying information, such as case IDs or patient IDs, is omitted during the aggregation process.

Apply differential privacy to trace variants. After constructing the tabular dataset $\mathbf{X}$ from the previous step, we limit the trace variants c_v to a (ε, δ)-differentially private selection of trace variants. This ensures that no information leakage occurs, as discussed in Section 3.2. We use the *TraVaS* algorithm proposed by [23] to obtain a set $\tilde{c_v}$ of (ε, δ)-differentially private trace variants. By removing all rows where the trace variant is not in the set $\tilde{c_v}$ from the tabular dataset $\mathbf{X}$, we obtain a new tabular dataset $\tilde{\mathbf{X}}$.

Apply tabular data generation algorithm. This step takes the tabular data set $\tilde{\mathbf{X}}$, applies a TDG $\mathcal{A}$ and returns the anonymised tabular dataset $\mathbf{X^P}$. As defined in Section 2.4, we denote the application of the TDG by $\mathbf{X^P} = \mathcal{A}(\tilde{\mathbf{X}})$. After this step, $\mathbf{X^P}$ satisfies (ε, δ)-DP, given that $\mathcal{A}$ guarantees (ε, δ)-DP. Note that any tabular data generation method that guarantees (ε, δ)-DP can be interchangeably used. Some methods additionally require the user to specify types for the columns in the dataset. This is abstracted in the framework and must be specified once in the beginning for each case attribute in the event log.

Rebuild event log. The generated tabular data can be transformed back into an event log format by creating a case for each row in the generated tabular dataset $\mathbf{X^P}$. For each row, a case is created that is annotated with the information from the case attributes and corresponding activity from the variant information. Based on the post-processing theorem of DP [4], the privacy guarantee of the anonymised dataset is preserved under any data transformation that does not involve additional queries to the original data. This means that any derived event log from L_p remains differentially private. Thus, improving the data utility after generation, e.g. by removing impossible combinations of case attributes, does not violate the privacy guarantee.

Calculating the final privacy budget. The framework takes as input two privacy budgets $(\varepsilon_{\text{TraVaS}}, \delta_{\text{TraVaS}})$ and $(\varepsilon_{\mathcal{A}}, \delta_{\mathcal{A}})$. Both budgets are used to ensure (ε, δ)-DP for the final event log. The output from the first mechanism, *TraVaS*, is used as input for the TDG. Therefore, instead of using the sequential composition theorem, the resulting composed privacy budget (ε, δ) needs to be calculated using the k-fold adaptive composition theorem as in Theorem 1.

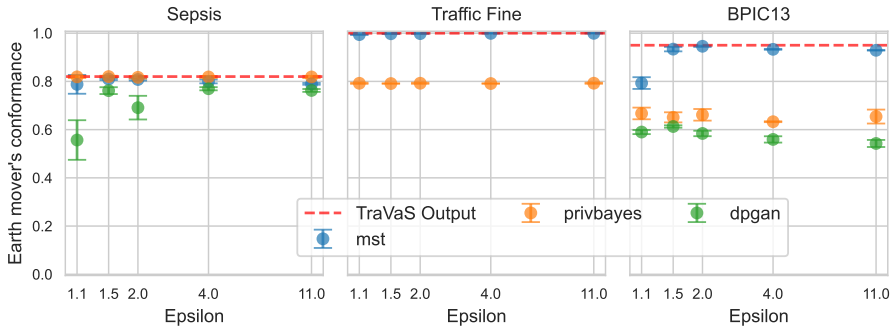

Fig. 3: Earth mover's conformance for values of ε.

4 Evaluation

This section evaluates the results of DP-ELCA when applying it to three real-life event logs using three different TDGs. The resulting anonymised event logs are evaluated regarding the similarity of the stochastic process behaviour, case attributes and the relationships within the event log to the original event logs. Further, each combination of tabular data generation is evaluated for different privacy budgets ($\varepsilon \in \{1.1, 2, 2.5, 4, 11\}$, $\delta = 0.75$. We set $(\varepsilon_{\text{TraVaS}}, \delta_{\text{TraVaS}}) = (0.1, 0.5)$ and $\tilde{\delta} = 0.0001$. Each combination of event log, TDG and privacy budget is run five times to account for the non-determinism of the TDG.

We choose *PrivBayes* [31], *MST* [18] and *DPGAN* [29] as TDGs, based on the availability of the implementation of each algorithm and the results in the literature [27]. For *PrivBayes* and *DPGAN* we use the implementation in the framework Synthcity [21]. *MST* is implemented in the framework Smartnoise-sdk github.com/opendp/smartnoise-sdk.

Stochastic process behaviour. We use the earth mover's conformance (EMC) [16] to measure the similarity of the stochastic process behaviour between the original and anonymised event log. This metric quantifies how closely the distribution of trace variants in the anonymised log matches the original distribution, using a value $d \in [0, 1]$. Figure 3 shows the results we obtained where an earth mover's conformance of 1 signifies perfect conformance. We observe a high conformance of the anonymised process behaviour to the original process behaviour. Further, *MST* produces better results for the traffic fine event log and the BPIC13 event log. The output from *TraVaS* is part of the input for the TDG algorithm in our framework. When comparing its EMC to the EMC of the TDG algorithms, we see only a slight decrease in conformance for the best TDG algorithms. Additional results for precision and fitness values are shown in [28].

Descriptive statistics of case attributes. Table 1 shows the statistical properties for the Sepsis event log for *MST*. We observe that for stricter privacy budgets the means and variances of the case attributes deviate more from the values of

MST						
ε	μ_{age}	$\mu_{infectionsuspected}$	$\mu_{hypotensie}$	$\mu_{infusion}$	$\mu_{oligurie}$	$\mu_{hypoxie}$
orig	70.08 ± (17.36)	0.81 ± (0.39)	0.05 ± (0.22)	0.76 ± (0.43)	0.02 ± (0.15)	0.02 ± (0.14)
11.0	66.77 ± (19.34)	0.67 ± (0.47)	0.03 ± (0.18)	0.62 ± (0.49)	0.02 ± (0.14)	0.02 ± (0.14)
4.0	65.25 ± (20.26)	0.67 ± (0.47)	0.03 ± (0.18)	0.61 ± (0.49)	0.04 ± (0.19)	0.02 ± (0.15)
2.0	55.64 ± (21.86)	0.67 ± (0.47)	0.05 ± (0.21)	0.62 ± (0.48)	0.04 ± (0.19)	0.04 ± (0.17)
1.5	55.66 ± (21.63)	0.70 ± (0.45)	0.05 ± (0.19)	0.62 ± (0.48)	0.05 ± (0.15)	0.02 ± (0.08)
1.1	55.27 ± (21.55)	0.51 ± (0.50)	0.50 ± (0.50)	0.50 ± (0.50)	0.51 ± (0.50)	0.40 ± (0.40)

Table 1: Means and standard deviations for different ε for Sepsis.

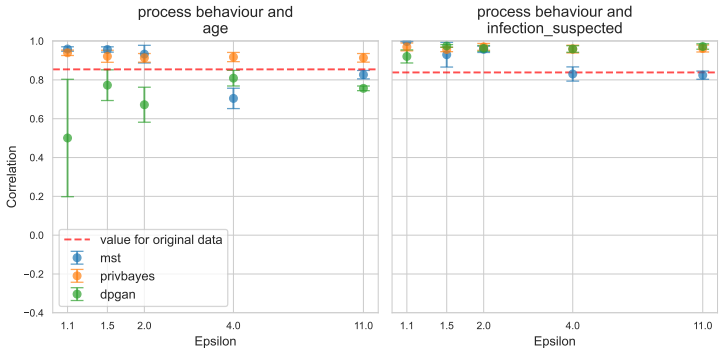

Fig. 4: Process behaviours' correlation with case attributes for Sepsis.

the original event log. *PrivBayes* and *DPGAN* do not show such a clear trend with the values, as shown in [28]. We find that the results for the different TDG algorithms do not differ significantly. However, in some cases, *DPGAN* fails to reproduce the distribution of binary case attributes.

Correlation process behaviour - case attributes. To test how our approach influences the relation between process behaviour and case attributes, we measure this correlation using [14] for several values of ϵ. Figure 4 shows the results for the Sepsis event logs, the other results are shown in [28]. We see that all TDG algorithms reproduce the existing correlations in the original event log. However, *MST* and *PrivBayes* produce more stable results than *DPGAN*, which deviates in some cases from the original values, especially for lower privacy budgets. Further, we see a higher variance in the results of *MST* for the lowest privacy budget.

Correlation between case attributes. We compare the Pearson correlation coefficient of the numerical case attributes between the original and anonymised event logs. Figure 5 shows the correlations between the case attributes of the traffic fine event log. We find that the *MST* algorithm underestimates the correlations, while the *PrivBayes* algorithm overestimates them. *DPGAN* is missing due its long run times for large event logs and computational limitations. The traffic fine event log is the only one of the chosen event logs containing multiple numerical case attributes.

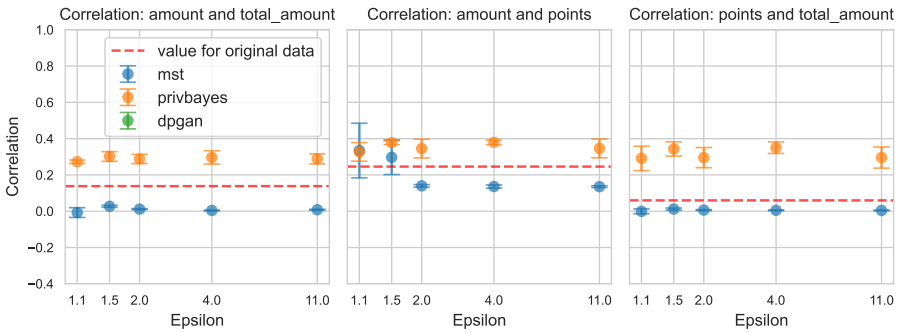

Fig. 5: Correlation measures: case attributes for Traffic fine.

5 Related Work

Several papers argue for privacy in process mining by highlighting the importance of protecting individuals' data [1,6]. A direction of research is to develop algorithms to provide group-based privacy guarantees, e.g. k-anonymity [26], for event logs [11,20,22]. These methods differ to our approach in the privacy guarantee given for the anonymised event log.

DP has been applied to anonymise event logs in process mining. Prefix-based methods utilise a noisy prefix tree to anonymise the trace variants [17]. This potentially introduces new, non-original trace variants. To limit this, the construction of the prefix tree can be guided using a score function that is derived from the usefulness of the possible trace variants [9]. Similarly, methods based on Deterministic Acyclic Finite State Automata (DAFSA) add noise to the transition frequencies [7]. This reduces the spent privacy budget while maintaining a high utility, which is achieved by combining the results of applying the algorithms to subsamples of the event log [5]. A method for trace variant anonymisation using a generative adversarial network has been proposed [24]. However, this approach does not filter out infrequent trace variants or modify them, thus posing re-identification risks, see Section 3.2. More recently, *TraVaS*, the adoption of a differentially private partition selection algorithm for trace variants, was proposed that enables the publication of unmodified trace variants where the noisy counts are above a certain threshold [23]. All of the before mentioned approaches to guarantee DP to event logs only include control flow information and no additional information in the anonymised event logs. Finally, Fahrenkrog-Petersen et al. [10] propose a two-step approach that anonymises control flow and adds contextual information. This method assumes the independence of case attributes, limiting its applicability to real-world logs. Our framework builds on the *TraVaS* approach, further enhancing the utility by using TDG to anonymise dependent case attributes.

6 Conclusion

This work proposed DP-ELCA, a differentially private framework, to anonymise event logs with case attributes ensuring (ε, δ)-DP while ensuring data quality. We make use of TDG algorithms that provide differential privacy for tabular data. Further we discuss privacy implications when using TDG algorithms on event logs and propose to filter the event log based on a differentially private set of trace variants obtained by using *TraVaS*. This allows for the release of the anonymised event logs with dependent case attributes. The k-fold adaptive composition theorem is used to compute the resulting privacy budget.

The evaluation of three real-life event logs shows that of the chosen TDG algorithms, the MST algorithm yields the best results in terms of the similarity of the anonymised event log to the original event log. In some cases, however, a trade-off between privacy and utility can be noticed. A lower privacy budget, i.e. a stronger privacy guarantee, decreases the similarity of the original and anonymised event log.

The framework's flexibility in the choice of the TDG algorithm ensures that future advancements in the field can be leveraged. Further evaluation with other TDG algorithms and systematic evaluation of hyperparameters could improve the framework's performance. Additionally, evaluation with other real-life event logs or controlled synthetic event logs could reveal further strengths and weaknesses. An avenue for future work lies in extending the framework to include additional dimensions, such as timestamps or more granular event attributes. Additionally, the task of choosing the right epsilon value balancing the trade off between privacy and utility could be investigated and methods to detect suitable privacy budgets developed.

References

1. van der Aalst, W., et al.: Process Mining Manifesto. In: Business Process Management Workshops. pp. 169–194. Springer (2012)
2. Cohen, A., Nissim, K.: Towards formalizing the GDPR's notion of singling out. Proceedings of the National Academy of Sciences **117**(15), 8344–8352 (2020)
3. Dwork, C.: Differential privacy. In: Automata, Languages and Programming. p. 1–12. Lecture Notes in Computer Science, Springer (2006)
4. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science **9**(3–4), 211–407 (2014)
5. Elkoumy, G., Dumas, M.: Libra: High-utility anonymization of event logs for process mining via subsampling. In: ICPM. pp. 144–151 (2022)
6. Elkoumy, G., Fahrenkrog-Petersen, S.A., Sani, M.F., Koschmider, A., Mannhardt, F., Von Voigt, S.N., Rafiei, M., Waldthausen, L.V.: Privacy and Confidentiality in Process Mining. ACM Trans. Manage. Inf. Syst. **13**(1), 1–17 (Mar 2022)
7. Elkoumy, G., Pankova, A., Dumas, M.: Mine me but don't single me out: Differentially private event logs for process mining. In: ICPM. pp. 80–87 (2021)
8. Elkoumy, G., Pankova, A., Dumas, M.: Differentially private release of event logs for process mining. Information Systems **115**, 102161 (2023)

9. Fahrenkog-Petersen, et al.: Sacofa: Semantics-aware control-flow anonymization for process mining. In: 2021 ICPM. pp. 72–79 (2021)
10. Fahrenkrog-Petersen, S.A., van der Aa, H., Weidlich, M.: PRIPEL: Privacy-Preserving Event Log Publishing Including Contextual Information. In: Business Process Management. pp. 111–128. Lecture Notes in Computer Science (2020)
11. Fahrenkrog-Petersen, S.A., et al.: Optimal event log sanitization for privacy-preserving process mining. Data Knowl. Eng. **145**, 102175 (2023)
12. Fung, B.C.M., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: A survey of recent developments. ACM Comput. Surv. **42**(4) (2010)
13. Kairouz, P., Oh, S., Viswanath, P.: The composition theorem for differential privacy. In: ICML. pp. 1376–1385 (2015)
14. Leemans, S.J.J., McGree, J.M., Polyvyanyy, A., ter Hofstede, A.H.: Statistical tests and association measures for business processes. IEEE Transactions on Knowledge and Data Engineering **35**(7), 7497–7511 (2023)
15. Leemans, S.J.J., Shabaninejad, S., Goel, K., Khosravi, H., Sadiq, S., Wynn, M.T.: Identifying Cohorts. In: Conceptual Modeling. pp. 92–102 (2020)
16. Leemans, S.J.J., et al.: Stochastic process mining: Earth movers' stochastic conformance. IS **102**, 101724 (2021)
17. Mannhardt, F., et al.: Privacy-preserving process mining: Differential privacy for event logs. Business & Information Systems Engineering **61**, 595–614 (2019)
18. Mckenna, R., Sheldon, D., Miklau, G.: Graphical-model based estimation and inference for differential privacy. In: ICML. vol. 97, pp. 4435–4444 (2019)
19. Márquez-Chamorro, A.E., Resinas, M., Ruiz-Cortés, A.: Predictive Monitoring of Business Processes: A Survey. IEEE Trans. on Serv. Comp. **11**(6), 962–977 (2018)
20. Pika, A., et al.: Privacy-preserving process mining in healthcare. Int. Jour. of Env. Res. and Pub. Health **17**(5) (2020)
21. Qian, Z., Cebere, B.C., van der Schaar, M.: Synthcity: facilitating innovative use cases of synthetic data in different data modalities (2023)
22. Rafiei, M., Wagner, M., van der Aalst, W.M.: TLKC-privacy model for process mining. In: Int. Conf. Res. Challenges Inf. Sci. pp. 398–416 (2020)
23. Rafiei, M., Wangelik, F., van der Aalst, W.M.P.: Travas: Differentially private trace variant selection for process mining. In: Proc. Min. Works. pp. 114–126 (2023)
24. Rafiei, M., Wangelik, F., Pourbafrani, M., van der Aalst, W.M.P.: TraVaG: Differentially Private Trace Variant Generation Using GANs. In: Res. Challenges Inf. Sci.: Inf. Sci. and the Connected World. pp. 415–431 (2023)
25. Sweeney, L.: Simple demographics often identify people uniquely. Health (San Francisco) **671**(2000), 1–34 (2000)
26. Sweeney, L.: k-anonymity: A model for protecting privacy. International journal of uncertainty, fuzziness and knowledge-based systems **10**(05), 557–570 (2002)
27. Tao, Y., McKenna, R., Hay, M., Machanavajjhala, A., Miklau, G.: Benchmarking differentially private synthetic data generation algorithms. arXiv (2021)
28. Ueck, H., Andrews, R., Wynn, M.T., Leemans, S.J.J.: Technical report: Differentially private event logs with case attributes, `https://github.com/hueck/DP-ELCA/blob/6cf7f578d3ee18f9cad2405fb25f0be0c43dd63b/technical_report.pdf`
29. Xie, L., Lin, K., Wang, S., Wang, F., Zhou, J.: Differentially private generative adversarial network (2018)
30. Yale, A., et al.: Generation and evaluation of privacy preserving synthetic health data. Neurocomputing **416**, 244–255 (2020)
31. Zhang, J., Cormode, G., Procopiuc, C.M., Srivastava, D., Xiao, X.: PrivBayes: Private data release via Bayesian networks. ACM Trans. Datab. Syst. **42**(4) (2017)