Filtering at the Edge: Exploring the Privacy-Utility Trade-Off

Maximilian Weisenseel[1], Fabian Sandkuhl[1], Henrik Kirchmann[2], Matthias Weidlich[2], and Florian Tschorsch[1]

[1] University of Technology Dresden
{maximilian.weisenseel,florian.tschorsch}@tu-dresden.de
fabian.sandkuhl@mailbox.tu-dresden.de
[2] Humboldt-Universität zu Berlin
{henrik.kirchmann,matthias.weidlich}@hu-berlin.de

Abstract. Process mining applications are no longer limited to traditional enterprise systems, but may involve data sensing at distributed event sources. However, when data is collected close to process stakeholders and processed immediately for online process control, privacy considerations become particularly important. For such scenarios, we argue that filtering of event data directly at the sources is a viable means to achieve certain privacy guarantees without drastically compromising the utility of the data. Specifically, we propose to leverage z-anonymity for distributed data filtering. It discards behavioral information that, due to it being infrequent, poses a high risk for re-identification attacks, but that is likely to be neglected as noise in downstream analysis tasks anyway. We explore the resulting interplay of data utility and privacy for different notions of behavioral information. Our experiments with several established event logs indicate that the non-linear dependencies between privacy and utility open up a space for effective data filtering.

Keywords: Privacy and Utility · Process Mining · Distributed Filtering

1 Introduction

During the past decade, process mining has been established as a means for the descriptive, predictive, and even prescriptive analysis of process-oriented information systems [1]. Recently, the adoption of process mining expanded beyond traditional enterprise systems, significantly extending its scope, in terms of the data used and the analysis performed [12]. Instead of exploiting a single centrally maintained event log for one-off retrospective analysis of a process, contemporary applications rely on data sensing close to process stakeholders and strive for a continuous and immediate assessment of a process' behavior [4].

Although event data enables valuable conclusions on the underlying process, it also carries privacy risks. That is, individuals may be re-identified based on their behavioral characteristics, e.g., through linkage with auxiliary data that is possessed by an adversary [17, 24]. To mitigate the respective risks, techniques for privacy-aware process mining received much attention recently [16,18].

Specifically, it has been shown how group-based privacy guarantees, such as k-anonymity and its derivatives, as well as differential privacy can be achieved for process-related event data [8, 9, 16, 19]. Yet, these techniques typically induce a trade-off: Stronger privacy guarantees come with lower data utility, i.e., the process mining results become less accurate or more uncertain.

While the privacy-utility trade-off is widely acknowledged [3], we note that common process mining applications include opportunities to manage this trade-off effectively. Process-related event data often contains outliers and infrequent behavior. Regardless of whether such behavior is genuine or stems from data quality issues, it is typically removed as part of data pre-processing [14, 21–23]. The reason being that many analysis techniques adopt a global, aggregated view on the process behavior, i.e., by discovering the behavioral characteristics that hold true for the vast majority of process executions. At the same time, suppression of events is also a common technique used to increase the privacy of the data [9]. Therefore, filtering of event data to achieve a privacy guarantee does not necessarily compromise the utility of the data for certain analysis tasks. Once the event data is captured in a distributed environment, the suppression of rare behavior is particularly valuable from a privacy point of view: If event data is filtered at the sources, i.e., directly at the edge of a distributed infrastructure, the respective behavior is not visible beyond the source. Thereby, the privacy strategies of *minimization* and *separation* are implemented by design [11].

In this paper, we build upon the aforementioned idea and present an approach to filter event data directly at the sources. Specifically, we propose to leverage *z-anonymity* [13], which discards rare behavioral information, suppressing behavior until at least $z - 1$ other individuals have shown it in the same timeframe. Furthermore, we introduce an adaptation, referred to as *explicit z-anonymity*, which publishes all behavior performed by at least z individuals in a specified timeframe. As such, instead of suppressing the $z - 1$ initial occurrences, *explicit z-anonymity* provides an explicit anonymity set. Either way, data that poses a high risk for re-identification attacks [24] is not released by the data source. Arguably, as discussed above, such data may be neglected in certain downstream analysis tasks in any case, so that an improvement of the analysis privacy may potentially be achieved without a drastic reduction of the data utility.

Against this background, we explore the privacy-utility trade-off achieved by our approach in a series of experiments. Using established event logs, such as Sepsis [15] and BPIC'12 [7], we assess the information loss induced by *(explicit) z-anonymity* on the level of individual traces, as well as for the event data as a whole. Moreover, we also assess the implications for process model discovery in terms of common evaluation dimensions. In general, our results demonstrate the feasibility and highlight the potential for effectively managing the privacy-utility trade-off due to the non-linear relations between privacy and utility measures.

In the remainder, Section 2 introduces our approach to distributed data filtering. Section 3 presents experimental results on the effectiveness of our approach. Section 4 reviews related work, before we conclude the paper in Section 5.

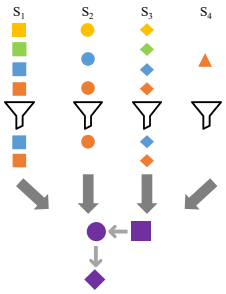

Fig. 1: Overview.

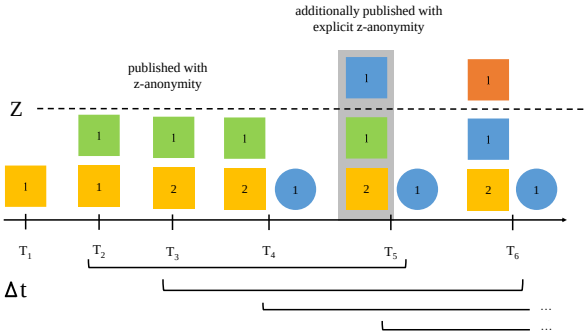

Fig. 2: Filtering with *z-anonymity*.

2 Filtering Event Data at Sources

In this section, we first give an overview of our approach to data filtering at the sources, before we turn to a formal definition of the respective privacy notions. In traditional process mining applications, data is assumed to be available as an event log at a central location [1]. This assumption fails to acknowledge that data is often collected in a distributed environment over time, with the desire to analyze it continuously. Taking into account that rare behavior of the process is often removed as part of data pre-processing, we consider a setting as illustrated in Figure 1: Data is continuously generated by different sources (S_1 to S_4), resulting in streams of events that signal activity executions for specific cases (activities are encoded by the shape type, while colors signify distinct cases). The data is then filtered locally, directly at the event sources, to not only clean the data from rare, supposedly noisy behavior, but to also safeguard privacy.

Specifically, we propose to apply *z-anonymity* [13] at the sources to filter the event streams. *z-anonymity* ensures that an event is only published if the same behavior was observed for at least $z-1$ other cases within a time frame Δt.

Figure 2 illustrates *z-anonymous* filtering for one source, when considering the executions of single activities as the relevant notion of behavior. Again, shape types and colors denote activities and cases, respectively, while the numbers indicate how often an activity was observed in a case. Consider the parameters $z = 3$ and $\Delta t = 4$. At time point T_1, the square activity is performed in the yellow case, followed by the green case at T_2, and, again, by the yellow case at T_3. Yet, since the yellow case has already been counted within the time frame, this occurrence does not increase the overall count. At T_4, a third case (blue) performs another activity, followed by the square activity at T_5. As $z = 3$ cases include the square activity within the specified time frame Δt, the blue square is published. At T_6, the event of the orange case results again in z cases, including the square activity within the time frame, so that the orange square is published.

We can observe that *z-anonymity* enables the publication of the information about the blue and the orange square activity, as z users have performed the

activity within the time frame Δt. However, the same statement also holds for the information about the executions of the square activity for the green and the yellow cases. Hence, we propose the concept of *explicit z-anonymity*, which publishes all information about behavior that has been performed at least z times in the time frame Δt. In our example, *explicit z-anonymity* would also publish the green and the yellow square, thereby making the anonymity set explicit.

Definition of Privacy Guarantees

We continue with a formal definition of *(explicit) z-anonymity* on event streams. Our model is based on sets of activities A, case IDs C, timestamps T, and payload data elements D. An event e is defined as a tuple consisting of an activity, a case ID, a timestamp, and a data element, i.e., $e = (a, c, t, d) \in \mathcal{E} = (A \times C \times T \times D)$. For the sake of simplicity (and without loss of generality), we restrict the presentation to a single data element. We use $e(X)$ to reference the attributes of event e, whereby $X \in \{A, C, T, D\}$.

Each data source in our setting generates a stream of events, which we model as a set of events $S \in 2^{\mathcal{E}}$. We assume that the events in a stream can be totally ordered by their timestamps. In our model, any source can generate any type of event; we do not impose restrictions that associate specific events, such as those of a particular activity, with specific sources.

Moreover, we consider different notions of behavior that may be observed in an event stream in the context of a particular case. We illustrate this definition with common behavioral features in process mining. Let E be a set of events. We model these behavioral features as Boolean predicates $\beta(E)$ that evaluate to true if the events in E show the respective behavior. The restriction on the set of admissible elements is used when applying the Boolean predicates in the *z-anonymity* definition. In this setting, E denotes a subset of events that satisfy this predicate and the *z-anonymity* properties, and hence is released.

- *Occurrences of activities*: If relevant behavior is induced by activity occurrences, β refers to the presence of an event signaling the execution of a specific activity $a \in A$:

$$\beta_a(E) \;\equiv\; \exists e \left(E = \{e\} \;\wedge\; e(A) = a\right).$$

- *Conditioned occurrences of activities*: The relevant behavior may be induced by activity occurrences that show certain payload data. For instance, only occurrences of an activity that took very long or that have been conducted by a specific resource may be privacy-sensitive. For a selected activity $a \in A$ and data element $d \in D$, the predicate is given as:

$$\beta_{a,d}(E) \;\equiv\; \exists e \left(E = \{e\} \;\wedge\; e(A) = a \;\wedge\; e(D) = d\right).$$

- *Sequences of occurrences of activities:* Relevant behavior may also include occurrences of activities that follow each other directly (with a pair of such occurrences representing the well-known directly-follows relation). For a se-

quence $\langle a_1, a_2, \ldots, a_n \rangle$ with $a_i \in A$, we define the respective predicate as:

$$\beta_{\langle a_1,\ldots,a_n\rangle}(E) \;\equiv\; \exists e_1, \ldots, e_n \Big(E = \{e_1, \ldots, e_n\} \;\wedge\; \bigwedge_{i=1}^{n} e_i(A) = a_i \\ \wedge\; e_1(T) < \cdots < e_n(T) \Big).$$

To define *z-anonymity*, we use auxiliary predicates on event sets $E, E' \in 2^{\mathcal{E}}$: $\mathcal{B}(E, E')$ asserts that E and E' exhibit the same behavior with respect to a chosen predicate β (possibly a conjunction of predicates capturing different behavioral features); $\mathcal{C}(E, E')$ asserts that they belong to distinct cases; $\mathcal{T}(E, E')$ asserts that the most recent events of E and E' lie in the same time window of length Δt; and $\mathcal{F}(E, E')$ asserts that the most recent event in E is more recent than the most recent event in E'.

$$\begin{aligned}
\mathcal{B}(E, E') &\equiv \beta(E) = \beta(E') \\
\mathcal{C}(E, E') &\equiv C_E \cap C_{E'} = \emptyset \wedge |C_E| = |C_{E'}| = 1 \\
&\qquad \text{with } C_E = \bigcup_{e \in E} \{e(C)\}, C_{E'} = \bigcup_{e \in E'} \{e(C)\} \\
\mathcal{T}(E, E') &\equiv \max_{e \in E} e(T) - \max_{e' \in E'} e'(T) \leq \Delta t \\
\mathcal{F}(E, E') &\equiv \max_{e \in E} e(T) > \max_{e' \in E'} e'(T)
\end{aligned}$$

Next, we formalize *z-anonymity* and *explicit z-anonymity* over an event stream S using the above predicates. Standard *z-anonymity* publishes an event $e \in S$ only if at least $z - 1$ other events feature the same relevant behavior as e, each belongs to a different case from one another and all belong to cases different from e's case, and all occurred within the time window Δt preceding e. If the behavioral predicate ranges over multiple events (e.g., $\beta_{\langle a_1,\ldots,a_n\rangle}$), then once the latest event in the set $E = \{e_1, \ldots, e_n\}$ occurs and the above conditions are satisfied with respect to that event, the entire set E is released (not just the triggering event). Based on these conditions, *z-anonymity* is formally defined as:

$$\text{zanon}(z, S) = \left\{ e \in S \;\middle|\; \begin{array}{l} \exists X \subseteq 2^S \Big(|X| = z \\ \quad \wedge\; \forall E, E' \in X \; (E \neq E' \Rightarrow \mathcal{C}(E, E')) \\ \quad \wedge\; \exists E \in X \; \big[e \in E \wedge\; \forall E' \in X \setminus \{E\} \; (\mathcal{B}(E, E') \\ \qquad \wedge \mathcal{T}(E, E') \wedge \mathcal{F}(E, E')) \big] \Big) \end{array} \right\}$$

Explicit z-anonymity extends the above notion: Whenever z distinct cases exhibit the same specified behavior within a given time window, we publish all events involved in that behavior from every case that contributed to reaching the z threshold, rather than only disclosing events belonging to cases at or above

that threshold. Formally, it is captured as:

$$\text{ezanon}(z, S) = \left\{ e \in S \;\middle|\; \begin{array}{l} \exists X \subseteq 2^S \; \Big(|X| = z \wedge \; \exists E \, (E \in X \wedge e \in E) \\ \qquad \wedge \; \forall E, E' \in X \; (E \neq E' \Rightarrow \mathcal{C}(E, E')) \\ \qquad \wedge \; \exists E \in X \; \big[\forall E' \in X \setminus \{E\} \; (\mathcal{B}(E, E') \\ \qquad\qquad \wedge \mathcal{T}(E, E') \wedge \mathcal{F}(E, E')) \big] \Big) \end{array} \right\}$$

Applying *(explicit) z-anonymity* at data sources realizes two fundamental design strategies for privacy notions [11]: *minimization* and *separation*. Through distributed filtering, the data propagation to the entity that is eventually responsible for process analysis is *minimized*. At the same time, outlying behavior remains confined to the location where it is recorded, which induces a *separation* of the respective sensitive data.

3 Empirical Evaluation

This section describes our experimental methodology and reports the results. We begin by outlining the evaluation setup, including datasets, anonymization configurations, and measured metrics. Then we present the outcomes of applying our approach, followed by a discussion of their implications. Our implementation and the used event data is available on GitHub.[3]

3.1 Setup

From Event Logs to Distributed Data Streams: For our evaluation, we use several real-world event logs that contain events with attributes that can be leveraged to simulate a distributed data streaming environment. In the Sepsis event log [15], we use the XES attribute `org:group`, defined in the XES standard as the organizational group to which the resource triggering the event belongs, to represent a partitioned source in the stream. In the other evaluated logs (the Environmental Permit log [5], the BPIC 2012 O log [7], and the BPIC 2020 Prepaid Travel Costs log [6]), we rely on the `org:resource` attribute, which specifies the particular resource, actor or system component, that generated the event. By treating each distinct group or resource as an originating stream, we model the event logs as a collection of distributed, concurrent data streams.

*Applying z-*anonymous *Filtering:* Given the individual streams induced by groups or resources from an event log, we apply the two variants of *z-anonymity*, *zanon* and *ezanon*, to each stream independently.

For each stream S, we apply the *z-anonymity* variant for the behavior predicate. We focus on behavioral predicates defined over sequences of activity occurrences, considering sequence lengths of 1, 2, and 3. The choice of time windows

[3] `https://github.com/henrikkirchmann/z_anonymity_pm.git`

is a challenging task that depends on the used log and the privacy and utility goals. The granularity of the recorded activities, as well as the concurrency of the cases, affects the number of event sequences that occur for distinct users and, therefore, the choice of a suitable time window. For example, a factory line that records a high number of identical activity sequences may be able to choose a small time window. While a hospital where activities are recorded by hand and thereby less often, and similar sequences are less likely, might require a larger time window. The choice of the time windows also affects the provided privacy, as the provided anonymity set depends on both the time window and the z value. In this paper, we focus on the effect of the z-values. Thereby, we fixed the time windows for all experiments to 72 hours and leave the choice of a suitable time window in relation to z as future work. For both variants, we sweep the anonymity parameter z from 1 to 30 to study how increasing the required group size affects privacy and utility. As an upper bound for comparison, we include a centralized, offline baseline filter on the same predicates. This baseline is an instantiation of *ezanon* with S set to the complete event log and the time window covering the entire log, thus omitting source partitioning and temporal restriction. It serves as a reference for the theoretically best trade-off between privacy and utility.

Evaluating Anonymized Data Streams: After each distributed stream induced from an event log L has been filtered, we reassemble the published events (preserving their original ordering within traces) into a new anonymized event log L'. Our goal is to quantify how much information and utility was lost through filtering, and how this trade-off impacts privacy. To this end we define the following utility metrics.

Ratio of Remaining Events and Traces. Let $|t|$ the number of events in a trace, and $|L|$ and $|L'|$ denote the number of traces in the original and anonymized logs, respectively. We define the ratio of remaining events (RRE) and the ratio of remaining traces (RRT) as

$$\mathrm{RRE} = \frac{\sum_{t' \in L'} |t'|}{\sum_{t \in L} |t|}, \qquad \mathrm{RRT} = \frac{|L'|}{|L|}.$$

Preservation of Directly-Follows Relations. To assess how much of the abstract behavior of the original process is retained, we consider the set of *directly-follows* relations, which is defined as:

$$\mathrm{DF}(L) = \{(a, b) \mid \exists \text{ trace } t \in L \text{ with activity } a \text{ immediately preceding } b \text{ in } t\}.$$

Then, the ratio of remaining directly-follows relations is

$$\mathrm{RDF} = \begin{cases} \frac{|\mathrm{DF}(L) \cap \mathrm{DF}(L')|}{|\mathrm{DF}(L)|} & \text{if } \mathrm{DF}(L) \neq \emptyset, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Fitness. To measure how much new or deviating behavior is introduced by filtering, we compute the *fitness* of the anonymized log L' against a process model M discovered from the original log L using the inductive miner. We use PM4Py's token based replay [2] to evaluate the fitness score. higher values indicate that the anonymized behavior is well explained by the original process structure, whereas lower values suggest the introduction of behavior not captured in L.

Re-identification Protection (A^* Projection). We measure identifiability by following the A^* projection in [25] checking how many traces are uniquely distinguishable when observing a small partial pattern of activities together with their day-granular timestamps. For each trace t, we sample

$$k = \left\lceil 0.1 \cdot \max_{t' \in L} |t'| \right\rceil$$

activity-timestamp pairs (with timestamps truncated to day granularity) and form the multiset pattern $S(t)$. Trace t is unique if no other trace $t' \neq t$ has full activity and day-level timestamp multisets that jointly contain $S(t)$ (i.e., $S(t)$ is not a multiset subset of any other trace). The risk is the fraction of such unique traces. To measure the provided protection, we calculate the complementary probability. Consequently, higher values indicate stronger protection against potential re-identification.

$$\text{Protection}_{A^*} = 1 - \frac{1}{|L|} \sum_{t \in L} \mathbf{1}[\nexists\, t' \neq t :\ S(t) \subseteq_{\text{multiset}} S(t')]\,.$$

3.2 Results and Discussion

We illustrated the results of our simulation in Figure 3. We initially focus on the ngram size one (first column), wherein sensitive behavior is defined as single activities. We can observe that as the value of z increases, both the number of events (first row) and remaining traces (second row) decrease; however, the degree of this reduction varies between the event logs. For instance, filtering the BPIC20 PTC log (red solid line) with $z = 5$, we observe a loss of nearly 20 percent of the events while retaining 95 percent of the traces. Furthermore, a value of $z = 10$ filters out 40 percent of the cases, yet it still retains nearly 90 percent of the traces. In contrast, the environmental permit log (blue solid line) exhibits a more substantial information loss, with 80 percent of the events filtered out at $z = 5$ and nearly 90 percent at $z = 10$. Consequently, the BPIC20 PTC log also retains more directly follows relations (row three) and a higher fitness score (row four). Furthermore, examining the re-identification protection (row five), indicates that higher z values also provide a stronger re-identification protection. The environmental permit log, which has the highest information loss by the considered metrics, is the one providing the best re-identification protection. The sharp declines observed in the Sepsis and environmental permit logs can be attributed to the minimal amount of retained data.

When behavior is defined using 2-grams (second column) or 3-grams (third column), we encounter significantly more information loss. This aligns with our intuition, as these finer-grained behavioral definitions raise the difficulty of satisfying the anonymity quota, resulting in a larger portion of data being discarded. Nevertheless, even for more challenging logs such as Sepsis (orange line), with 2-grams (second column) and $z = 5$, approximately half of the event information remains, indicating that general analysis insights (e.g., the primary control flow) can still be recovered despite the information loss. In general, we can observe a similar trend to the 1-grams. A higher z-value is correlated with a higher information loss and a higher re-identification protection. We can also observe a similar trend for the event logs. Also for 2- and 3-grams, the BPIC 2020 PTC log (red solid line) loses less information but offers less re-identification protection than the environmental permit log (blue solid line). When examining the preservation of the directly-follows relation, we find that filtering of 2-grams and 3-grams has a substantial impact. When comparing *z-anonymity* (solid lines) to *explicit z-anonymity* (dashed lines), we can observe that *z-anonymity* has a higher information loss across all four considered metrics, but is, in return, able to provide a better re-identification protection. The largest difference is in the ratio of remaining events, which is expected: *z-anonymity* suppresses more events than *explicit z-anonymity*. Note, however, that our re-identification protection metric understates the effect of non-explicit anonymization, since it does not account for deleted events; this can lead to more unique traces in the anonymized log than the metric suggests. Finally, analyzing the interplay between information loss and re-identification risk, we find that the BPIC 12 O log (to a lesser degree in the Sepsis log) demonstrates the ability to strengthen the privacy guarantee without degrading utility, which suggests a favorable privacy-utility trade-off.

4 Related Work

Privacy-aware process mining received much attention recently. As mentioned, the respective techniques adopt different notions of privacy, including group-based privacy guarantees [9,19], such as k-anonymity, t-closeness, and l-diversity, and differential privacy [8,16]. By adopting and adapting *z-anonymity*, our work belongs to the first group of techniques, as it has been shown that *z-anonymity* can provide k-anonymity with a desired probability [13]. However, unlike the existing techniques that focus on the sanitization of event data that is available at a single location, our work shows how to leverage the distribution of event sources as encountered in scenarios involving data sensing. Group-based privacy guarantees may be achieved through different types of data transformations, i.e., data suppression, data aggregation, or data generalization. While our approach suppresses event data, existing ideas on aggregation of process-related data (e.g., by merging traces [9]) or its generalization (e.g., by generalizing activities [10]) may also be incorporated in our distributed setting. Finally, as we target immediate processing of streaming data, we note that privacy risks induced by continuous data release have largely been ignored in process mining.

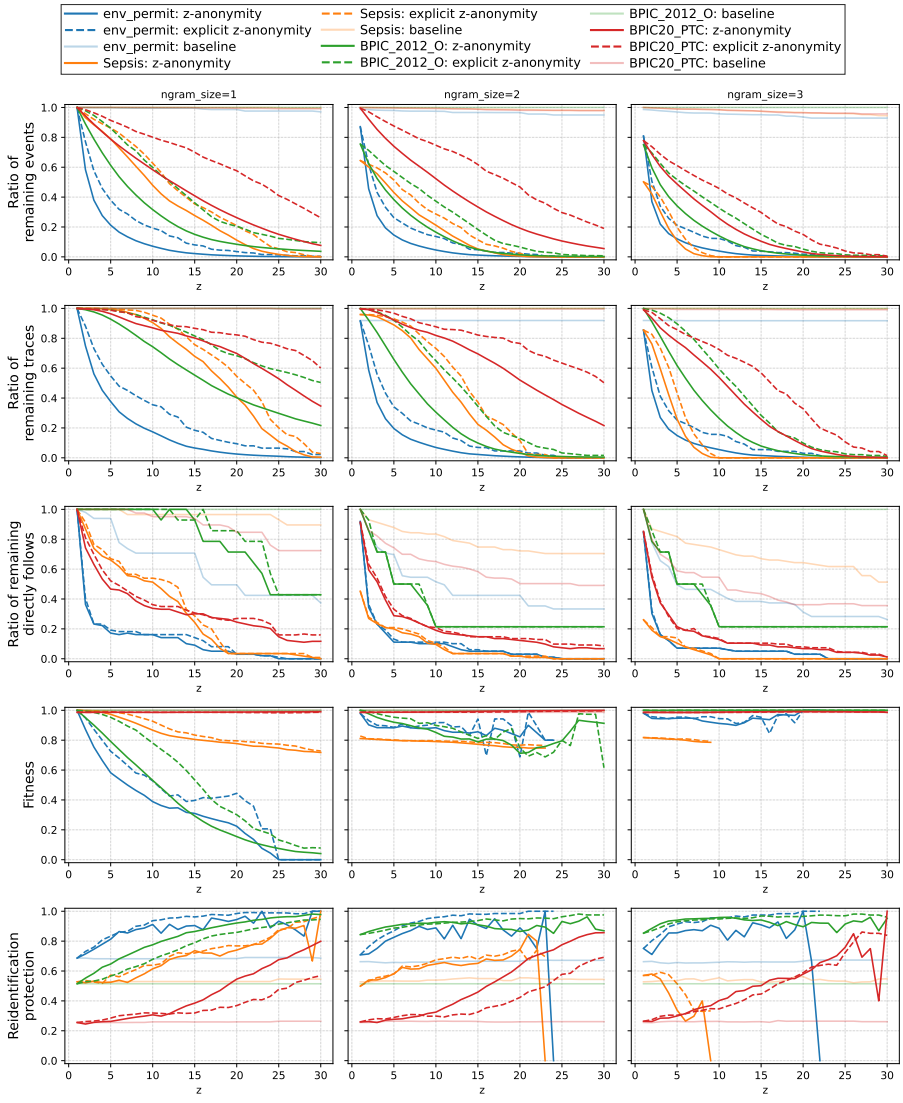

Fig. 3: Utility and privacy metrics for various logs and anonymization configurations. Higher values are better.

A notable exception has been the definition of correspondence attacks based on a comparison of releases [20]. However, similar risks have been studied in event stream processing. For instance, it has been shown how the suppression of events in a stream prevents the evaluation of sensitive patterns [26]. We see such pattern-specific techniques as a promising extension of our work, in cases where knowledge on sensitive behavioral patterns is available.

5 Conclusion

In this paper, we presented an approach for distributed filtering of event data directly at event sources as a foundation for privacy-aware process mining. Specifically, we showed how *(explicit) z-anonymity* can be achieved for notions of behavior that are commonly encountered in process analysis. We further argued that filtering of rare behavior at event sources provides an opportunity to effectively manage the privacy-utility trade-off, as the behavior filtered for reasons of privacy is likely to be neglected as noise in downstream analysis tasks. Indeed, our experimental results with several established event logs illustrate how a certain level of privacy may be reached, for some datasets and configurations, without incurring a large information loss. Our results open up various directions for future work. First, the observed differences in the results obtained for different datasets raise the question of how to predict the resulting utility when increasing the strength of a privacy guarantee. Furthermore, the effect of *z-anonymity* on utility dimensions of precision, generality, and simplicity seems promising. In addition, the effect of the temporal dimension, i.e., the time window adopted in *z-anonymity*, requires further exploration. Finally, we aim at further support for achieving Pareto optimality for privacy and utility in a specific setting.

Acknowledgments. This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – FOR 5495.

References

1. van der Aalst, W.M.P.: Process mining: A 360 degree overview. In: Process Mining Handbook, LNBIP, vol. 448, pp. 3–34. Springer (2022)
2. Berti, A., van Zelst, S.J., Schuster, D.: Pm4py: A process mining library for python. Softw. Impacts **17**, 100556 (2023)
3. Brickell, J., Shmatikov, V.: The cost of privacy: destruction of data-mining utility in anonymized data publishing. In: KDD. pp. 70–78. ACM (2008)
4. Brzychczy, E., Aleknonyte-Resch, M., Janssen, D., Koschmider, A.: Process mining on sensor data: a review of related works. Knowl. Inf. Syst. **67**(6), 4915–4948 (2025)
5. Buijs, J.: Receipt phase of an environmental permit application process ('wabo'), coselog project. Eindhoven University of Technology (2014)
6. van Dongen, B.: Bpi challenge 2020 (2020). `https://data.4tu.nl/collections/BPI_Challenge_2020/5065541/1`
7. van Dongen, B.: Bpi challenge 2012 (2012). https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f

8. Elkoumy, G., Pankova, A., Dumas, M.: Differentially private release of event logs for process mining. Inf. Syst. **115**, 102161 (2023)
9. Fahrenkrog-Petersen, S.A., van der Aa, H., Weidlich, M.: Optimal event log sanitization for privacy-preserving process mining. DKE **145**, 102175 (2023)
10. Hildebrant, R., Fahrenkrog-Petersen, S.A., Weidlich, M., Ren, S.: PMDG: privacy for multi-perspective process mining through data generalization. In: CAiSE. LNCS, vol. 13901, pp. 506–521. Springer (2023)
11. Hoepman, J.: Privacy design strategies. In: SEC. IFIP Advances in Information and Communication Technology, vol. 428, pp. 446–459. Springer (2014)
12. Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., Di Ciccio, C., Fortino, G., Gal, A., Kannengiesser, U., Leotta, F., et al.: The internet of things meets business process management: a manifesto. IEEE Systems, Man, and Cybernetics Magazine **6**(4), 34–44 (2020)
13. Jha, N., Favale, T., Vassio, L., Trevisan, M., Mellia, M.: z-anonymity: Zero-delay anonymization for data streams. In: IEEE BigData. pp. 3996–4005. IEEE (2020)
14. Koschmider, A., Kaczmarek, K., Krause, M., van Zelst, S.J.: Demystifying noise and outliers in event logs: Review and future directions. In: BPM Workshops. LNBIP, vol. 436, pp. 123–135. Springer (2021)
15. Mannhardt, F.: Sepsis cases - event log (2016). https://doi.org/10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460
16. Mannhardt, F., Koschmider, A., Baracaldo, N., Weidlich, M., Michael, J.: Privacy-preserving process mining - differential privacy for event logs. Bus. Inf. Syst. Eng. **61**(5), 595–614 (2019)
17. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: SP. pp. 111–125. IEEE Computer Society (2008)
18. Pika, A., Wynn, M.T., Budiono, S., ter Hofstede, A.H.M., van der Aalst, W.M.P., Reijers, H.A.: Towards privacy-preserving process mining in healthcare. In: BPM Workshops. LNBIP, vol. 362, pp. 483–495. Springer (2019)
19. Rafiei, M., van der Aalst, W.M.P.: Group-based privacy preservation techniques for process mining. Data Knowl. Eng. **134**, 101908 (2021)
20. Rafiei, M., van der Aalst, W.M.P.: Privacy-preserving continuous event data publishing. In: BPM Forum. LNBIP, vol. 427, pp. 178–194. Springer (2021)
21. Sani, M.F., van Zelst, S.J., van der Aalst, W.M.P.: Applying sequence mining for outlier detection in process mining. In: OTM Conferences (2). LNCS, vol. 11230, pp. 98–116. Springer (2018)
22. Sun, X., Hou, W., Yu, D., Wang, J., Pan, J.: Filtering out noise logs for process modelling based on event dependency. In: ICWS. pp. 388–392. IEEE (2019)
23. Suriadi, S., Andrews, R., ter Hofstede, A.H.M., Wynn, M.T.: Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. Inf. Syst. **64**, 132–150 (2017)
24. 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. LNCS, vol. 12127, pp. 252–267. Springer (2020)
25. 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: Empiricial evaluation paper. In: Advanced Information Systems Engineering. vol. 12127, p. 252 (2020)
26. Wang, D., He, Y., Rundensteiner, E.A., Naughton, J.F.: Utility-maximizing event stream suppression. In: SIGMOD Conference. pp. 589–600. ACM (2013)