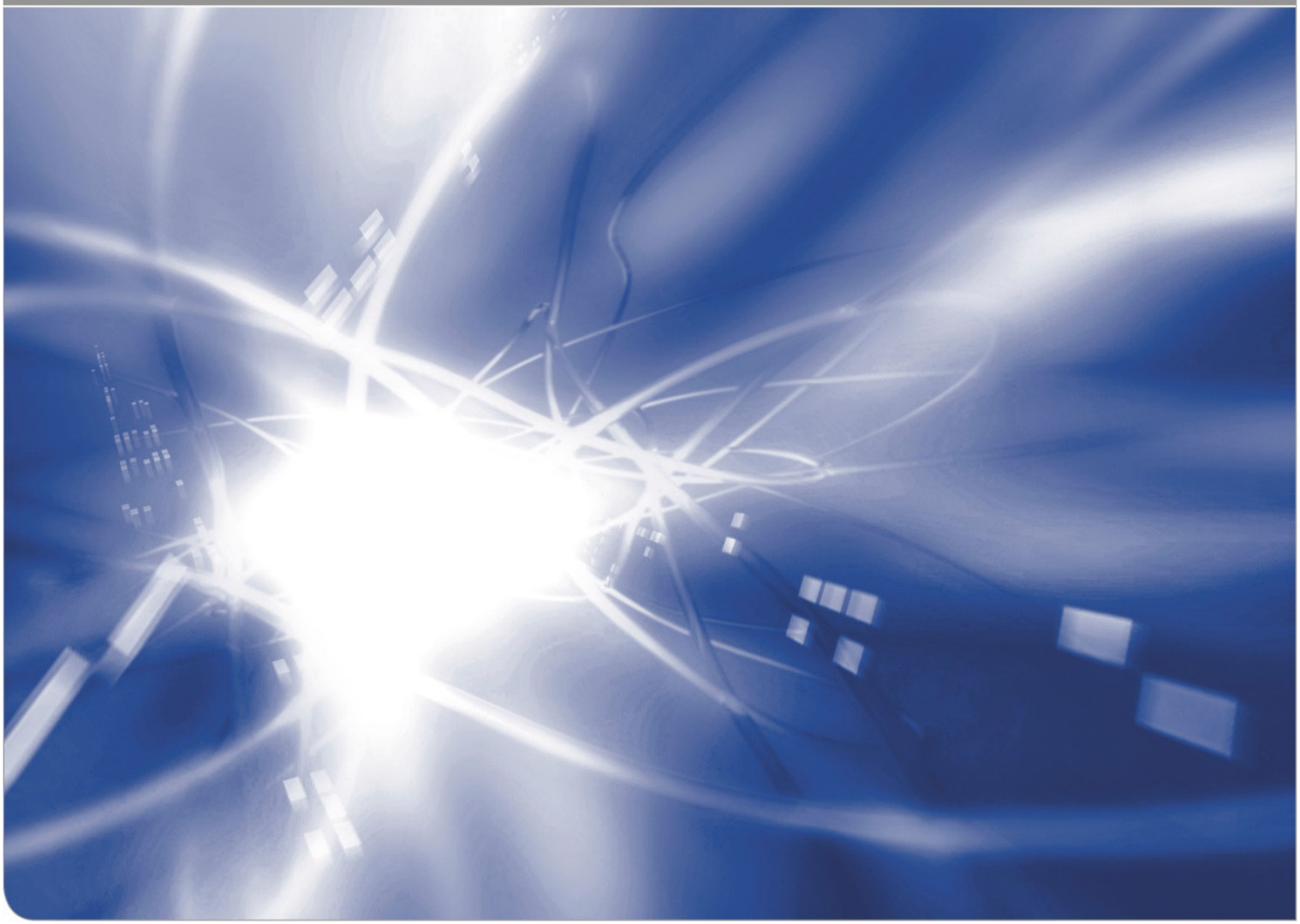
Karlsruhe Institute of Technology

Benchmarking the Utility of w-event Differential Privacy Mechanisms

– When Baselines Become Mighty Competitors

by Christine Schäler[1], Thomas Hütter[2], Martin Schäler[2]

KIT SCIENTIFIC WORKING PAPERS 194



[1] Institute for Program Structures and Data Organization
[2] University of Salzburg

Impressum

Karlsruher Institut für Technologie (KIT)
www.kit.edu

2022

ISSN: 2194-1629

Benchmarking the Utility of w-event Differential Privacy Mechanisms – When Baselines Become Mighty Competitors

Christine Schäler
Karlsruhe Institute of Technology
Karlsruhe, Germany
christine.schaeler@kit.edu

Thomas Hütter
University of Salzburg
Salzburg, Austria
thomas.huetter@plus.ac.at

Martin Schäler
University of University
Salzburg, Austria
martin.schaeler@plus.ac.at

ABSTRACT

The w-event framework is the current standard for ensuring differential privacy on continuously monitored data streams. Following the proposition of w-event differential privacy, various mechanisms to implement the framework were proposed. Their comparability in empirical studies is vital for both practitioners to choose a suitable mechanism and researchers to identify current limitations and propose novel mechanisms. By conducting a literature survey, we observe that the results of existing studies are hardly comparable and partially intrinsically inconsistent.

To this end, we formalize an empirical study of w-event mechanisms by a four-tuple containing re-occurring elements found in our survey. We introduce requirements on these elements that ensure the comparability of experimental results. Moreover, we propose a benchmark that meets all requirements and establishes a new way to evaluate existing and newly proposed mechanisms. Conducting a large-scale empirical study, we gain valuable new insights into the strengths and weaknesses of existing mechanisms. An unexpected – yet explainable – result is a baseline supremacy, i.e., using one of the two baseline mechanisms is expected to deliver good or even the best utility. Finally, we provide guidelines for practitioners to select suitable mechanisms and improvement options for researchers to break the baseline supremacy.

1 INTRODUCTION

Monitoring data streams continuously facilitates numerous new applications, e.g., controlling real-time intelligent traffic [17] or electricity distribution systems [2]. However, the privacy requirements of the data owner have to be fulfilled in order to deploy them. To ensure strong privacy for streams, the w-event differential privacy (DP) framework [20] is the standard state-of-the-art. The idea is to give a provable statistical indistinguishable guarantee of continuously calculated query results. The guarantee holds for any rolling window of at most w timestamps.

In the literature, various mechanisms are proposed that sanitize streams to achieve w-event differential privacy [7, 8, 21, 23, 26, 28]. All of these mechanisms sanitize the stream by injecting noise into the query results. Consequently, the design goal of such mechanisms is to minimize the introduced error and hence provide high data utility. Existing mechanisms aim to achieve high data utility by exploiting stream properties, e.g., sparse streams [28]. Unfortunately, there are little insights on which stream properties provide high data utility mainly due to incomparable empirical studies. This imposes a particular challenge for data administrators that need to choose a mechanism with suitable utility as well as researchers that aim to identify the utility limitations of existing solutions since theoretical examinations mostly analyze worst case scenarios.

Currently, there is no generally accepted and unified procedure to perform empirical studies on w-event DP mechanisms for streams. Quite the contrary, our literature survey reveals that existing studies significantly deviate in relevant aspects, e.g., input data streams and competitor mechanisms. This hampers the comparison and evaluation of existing results. Unfortunately, guidelines for empirical studies on static data [18] (e.g., finite time series [16] or relational databases [5]) cannot be applied since w-event mechanisms work significantly different. For example, rolling window techniques keep track of the available privacy budget for each window of size w. Summarizing, incomparable empirical studies limit the practical application of w-event mechanisms and delays the introduction of novel mechanisms in the research community.

Limitations of Existing Studies. The comparability of empirical studies on w-event DP is limited by inconsistent experimental elements. For example, the selection of data streams and competitor mechanisms, or the interpretation of the computed errors indicating a mechanism's utility.

Specifically, most studies focus on a small set of varying real-world data streams [7, 8, 21, 23, 26, 28]. We observe two challenges: First, the selection is hampered by the fact that many streams are not publicly available. Second, in case of available streams, most studies apply necessary preprocessing steps. However, the preprocessing might not be known [6, 24] and highly differs among publications using the same streams. Studies that use artificial data to investigate the influence of relevant data properties are only available for static data [18] and finite time series [16].

Analyzing available streams indicates that they are often sparse, i.e., they mainly contain zero values, especially multi-dimensional streams. Thus, publishing the same value all the time, as performed by one of the baseline mechanisms [20], yields good utility w.r.t. common error metrics.

Next, quantifying the benefit that data administrators can achieve from the latest w-event mechanism is virtually impossible since many studies do not compare to both baselines mechanism. Therefore, it is hard to decide whether an easy-implementable baseline suffices the use case, or whether a sophisticated mechanism is needed. Moreover, state-of-the-art mechanisms are highly complex and subtle differences in the implementation or initialization parameters can have a significant effect on a mechanism's utility. This is a serious limitation, especially since implementations of mechanisms are rarely publicly available.

Contributions. Motivated by the illustrated limitations of previous experimental studies, we present the following contributions:

Identification of benchmark requirements. Based on a comprehensive literature survey, we identify that all existing empirical

Table 1: Illustration of a location monitoring use case. Database D_t contains the location of all individuals at timestamp t. Query $Q(D_t)$ contains the number of individuals (cnt) per location. The goal is to hide trajectories of w timestamps.

Ind.	D_1	D_2	D_3	...
Axl	park	beach	park	...
Joan	park	park	beach	...
Rene	beach	beach	park	...
Query	cnt(park) = 2	cnt(park) = 1	cnt(park) = 2	...
$Q(D_t)$	cnt(beach) = 1	cnt(beach) = 2	cnt(beach) = 1	...

studies on w-event mechanisms can be described by four elements: mechanisms, streams, privacy requirements, and utility metrics. We outline the limitations of prior studies for each element, and propose and justify requirements on these elements to ensure the comparability of results. Moreover, our survey reveals that all existing w-event mechanisms follow the same abstract framework simplifying the comparison at a qualitative level.

Benchmark instantiation. We show how to meet the identified requirements and introduce the first benchmark for w-event DP. We include an artificial data generator that allows to analyze the influence of stream properties on a mechanism's utility. The benchmark establishes a new and comparable way to evaluate existing or newly proposed mechanisms and is publicly available[1].

Empirical study and new insights. We conduct the largest empirical evaluation of w-event DP mechanisms so far based on our benchmark, comprising of 252,000 single experiments. The results yield three main insights: Analyzing the influence of stream properties on a mechanism's utility, the amplitude is decisive rather than the period length. Further, an unexpected baseline supremacy is observed, i.e., one of the two baseline mechanisms provide the highest utility for every combination of stream and privacy requirements. Finally, data-adaptive sampling techniques do not yield a utility improvement if the amplitudes of the stream are large.

Discussion of takeaways. Considering the experimental results, we provide guidelines that help practitioners to select a suitable mechanism and reveal research directions for future work.

2 PRELIMINARIES

We start by providing required background knowledge on w-event differential privacy, the w-event mechanism framework for mechanism design, and introduce common utility metrics.

2.1 w-Event Differential Privacy

The w-event differential privacy is the current standard for ensuring differential privacy of aggregation queries computed on continuously monitored data streams. Rather than protecting the stream entirely which requires an infinite amount of noise, the key idea is to protect every running window of at most w timestamps [20].

Let $S = (D_1, D_2, ..)$ be a data stream collecting database D_t at timestamp t as shown in the example in Table 1. Each row in D_t corresponds to a individual and each column to an activity, i.e., location visit. For such a stream, a query of interest $Q(D_t)$ is the number of individuals per location at each timestamp. This query is computed using a multi-dimensional count query (i.e., histogram) with one count per location and timestamp. All differential privacy frameworks are built upon a notion of neighborhood, i.e., query results over a stream that are hardly distinguishable by an attacker. Two databases D_t, D'_t are neighbors if one can be obtained from the other by adding or removing one row, i.e., individual. Further, let $S_p = (D_1, \ldots, D_p)$ be a stream prefix of length p. Intuitively, two stream prefixes are w-neighbors if (1) the databases collected at each timestamp are pairwise the same or neighbors, and (2) all neighboring databases fit in a window of size w (cf. Definition 1).

Definition 1 (w-Neighboring Stream Prefixes [20]). *Let w be a positive integer, and $t, t_1, t_2 \leq p$ three timestamps. Two stream prefixes S_p, S'_p are* w-neighboring *if*

(1) D_t, D'_t are neighboring for each D_t, D'_t with $D_t \neq D'_t$
(2) $t_2 - t_1 < w$ for each $D_{t_1}, D_{t_2}, D'_{t_1}, D'_{t_2}$ with $t_1 < t_2, D_{t_1} \neq D'_{t_1}$ and $D_{t_2} \neq D'_{t_2}$.

The desired *privacy level* ϵ is set by the data administrator and usually lies between 0.1 and 1. A smaller value means better privacy. From Definition 2, w-event differential privacy is given if the query results of all w-neighboring stream prefixes are hard to distinguish ,i.e., up to a factor of e^{ϵ}.

Definition 2 (w-Event ϵ-Differential Privacy [20]). *Let $\mathcal{M}$ be a randomized mechanism that takes a stream prefix of arbitrary size as input. We say that $\mathcal{M}$ satisfies* w-event ϵ-differential privacy *if for all $R \in Range(\mathcal{M})$, all w-neighboring stream prefixes S_p, S'_p, and all p, holds that*

$$Pr[\mathcal{M}(S_p) = R] \leq e^{\epsilon} \cdot Pr[\mathcal{M}(S'_p) = R].$$

To implement a w-event DP mechanism for numeric queries, a mechanism usually adds noise based on the zero-mean Laplace distribution $\text{Lap}(\lambda)$ to each of the dim outputs of a query $Q : D \rightarrow \mathbb{R}^{\text{dim}}$, e.g., histogram bins. The scale $\lambda = \frac{\Delta Q}{\epsilon}$ depends on the privacy budget ϵ and the *global sensitivity* $\Delta Q = \max_{D,D'} ||Q(D) - Q(D')||_1$. The global sensitivity quantifies the maximum difference query results of neighboring databases may have. For instance, $\Delta Q = 1$ holds for a histogram query. Specifically, w-event DP can be implemented by using independent DP sub-mechanisms $\mathcal{M}_t$, e.g., Laplace mechanisms, to release the query results at a timestamp, as long as we ensure that the budget spend by these mechanisms does not exceed ϵ for every rolling window of size w (cf. Theorem 1).

Theorem 1 (Composition [20]). *Let $\mathcal{M}$ be a mechanism processing a stream prefix $S_p = (D_1, \ldots, D_p)$, and outputting a transcript of released values $R = (r_1, \ldots, r_p)$. Assume that we can decompose $\mathcal{M}$ into p sub-mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_p$, s.t. $\mathcal{M}_t(D_t) = r_t$, each $\mathcal{M}_t$ has independent randomness and achieves ϵ_t- differential privacy. Then, $\mathcal{M}$ satisfies w-event differential privacy if*

$$\forall t \in [w, p] : \sum_{k=t-w+1}^{t} \epsilon_k \leq \epsilon.$$

[1] https://dbresearch.uni-salzburg.at/projects/dpbench/index.html

2.2 The w-event Mechanism Framework

We now introduce an abstract framework for sub-mechanisms $\mathcal{M}_t$ (cf. Algorithm 1) that is suitable for all mechanisms of our literature survey. This common framework facilitates the comparison of mechanisms and experimental results.

Algorithm 1 w-event Mechanism Framework

```
1:  function M_t(ε, w, D_t, l)
2:      if isSamplingPoint(ε, w, D_t, l) then
3:          ε_t ← budgetAllocation(ε, w, D_t, l)
4:          p_t ← pertubation(ε_t, ΔQ, D_t)
5:          r_t ← filtering(p_t)              ▷ sanitized query result
6:          l ← t
7:      else r_t ← r_l                        ▷ approximation
8:      end if
9:      return r_t
10: end function
```

A sub-mechanism $\mathcal{M}_t$ has four inputs[2]: privacy requirements ϵ and w, database D_t, and the last timestamp l where a sub-mechanism released a *sanitized* query result. Note that not all timestamps of the query result are sanitized. Instead, previously sanitized query results can be released multiple times. The output of the sub-mechanism is the released query result r_t. Intrinsically, the sub-mechanism implements four functions that are described below. For illustration, Table 2 provides example implementations of these functions.

isSamplingPoint-Function. A mechanism has to decide whether a timestamp is sampled, i.e., the current query result is sanitized by spending a portion of the privacy budget ϵ for perturbation. Then, $\mathcal{M}_t$ releases this sanitized query result. The alternative to sampling is called *approximation*, i.e., the mechanism approximates the current query result with the one(s) sanitized last at timestamp l. The rationale for approximation is to save budget in case the query results change only marginally over time.

budgetAllocation-Function. This function is called in case the mechanism decides to sample. It determines and allocates the share of privacy budget used for perturbation. Here, the used strategies differ highly among the mechanisms.

Perturbation-Function. The mechanism first calculates the true query result which is then perturbed using the allocated budget. Note that all identified mechanisms leverage the Laplace mechanism for perturbation.

Filtering-Function. The post-processing immunity of differential privacy [12] allows to modify the perturbed query results p_t in an arbitrary way without spending budget or loosing the privacy guarantee, as long as no private information computed on D_t is used. Consequently, sub-mechanisms take advantage of this property within the filtering function to increase the utility. A straight-forward filtering function truncates the perturbed query result such that it fits in the domain Range(Q) of the query Q. For instance, Range(Q) contains all non-negative integers for count queries. During truncating, the mechanism takes the perturbed query result p_t and releases $max(0, \text{round}(p_t))$, where round is a function that rounds a floating point number to the next integer.

[2]Note that individual mechanisms may use additional input parameters.

Table 2: Computation of the functions in the w-event mechanism framework for the baselines Uniform and Sample [20].

Function	Uniform	Sample
isSamplingPoint	true	**if** w%t=0 **then** true **else** false
budgetAllocation	$\epsilon_t \leftarrow \frac{\epsilon}{w}$	$\epsilon_t \leftarrow \epsilon$
pertubation	$p_t \leftarrow Q(D_t) + \text{Lap}(\frac{\Delta Q}{\epsilon_t})$	
filtering	p_t	

2.3 Utility Metrics

To measure the utility of the released stream, researchers frequently quantify the difference of r_t to the true query result $Q(D_t)$ at each timestamp t mainly using the *mean absolute error (MAE)* or the *mean relative error (MRE)* [4, 15, 20, 21, 27, 30]. The mean absolute error is defined as

$$\text{MAE}(Q(S_p), R) = \frac{1}{p}\sum_{t=1}^{p} |Q(D_t) - r_t|.$$

Similar, for $\gamma > 0$, the mean relative error is defined as

$$\text{MRE}(Q(S_p), R) = \frac{1}{p}\sum_{t=1}^{p} \frac{|Q(D_t) - r_t|}{\max\{Q(D_t), \gamma\}}.$$

Here, γ is a sanity bound to mitigate the effect of small query results.

3 BENCHMARK REQUIREMENTS

In this section, we state and justify requirements on common elements of empirical studies on w-event mechanisms to ensure the comparability of their results. We identify these elements by conducting a comprehensive literature survey with the following methodology: We consider all publications that cite the original work on w-event DP [20]. We further include publications from the proceedings of the VLDB and SIGMOD conferences (2020 to 2022) and the ACM CCS conference (2020 to 2021). Summarizing, we include all publications that perform an experimental evaluation on *streams* (i.e., not only finite time series which excludes, e.g., [1, 10]) and are published at notable peer-reviewed conferences or journals. Note that this also includes publications on event-level DP or custom privacy definitions that generalize w-event DP. In total, we included 16 publications listed in Table 3.

As a result of our survey, we formalize the requirements of an empirical study on w-event mechanisms with a 4-tuple $(\mathbb{M}, \mathbb{S}, \mathbb{P}, \mathbb{E})$ with the following semantics:

- $\mathbb{M}$ is a set of mechanisms compared.
- $\mathbb{S}$ is a set of streams, i.e., datasets.
- $\mathbb{P}$ is a set of privacy requirements, i.e., (w,ϵ)-tuples.
- $\mathbb{E}$ is a set of (error) metrics to quantify mechanism utility.

We next describe the elements in more detail and introduce requirements on the elements that ensure the comparability of empirical studies. Based on the requirements, we reveal the limitations of existing studies.

Table 3: Requirements analysis of related work w.r.t. $\mathbb{M}$, $\mathbb{P}$, and $\mathbb{E}$ (✓yes, ✗no, ✓partially, - not considered). u denotes unknown and d dimension. Note that ($\mathbb{M}$-R4) and ($\mathbb{M}$-R5) are not applicable.

Reference	Privacy Definition	($\mathbb{M}$-R1) Proof	($\mathbb{M}$-R2) Baselines	($\mathbb{M}$-R3) Sources	($\mathbb{P}$ - R1) (ϵ, w)	($\mathbb{P}$ - R2) (w, ϵ)	($\mathbb{E}$ - R) Utility metrics
BA, BD, FAST$_w$ [20]	w-event	✓	both	✓[a]	-	([40,200], 1)	MAE, MRE $\gamma = u$
Retroactive Grouping [5]	event-level	✓	Uniform	✗	([000.2,0.1], 1)	n.a.	MAE, MRE $\gamma = u$
DSAT$_w$ [21]	w-event	✓	none[b]	✗	([0.5,1], 800)	([200,1000], u)	total sum of squared error
SecWeb [26]	w-event	✓	Uniform	✗	([0.01,1], 120)	([40,240], 1)	MAE, MRE $\gamma = 1$
G-event [8]	w-event	✗	Sample	✗	([0.5,1.5], u)	([40,200], u)	MAE, MRE $\gamma_d = 0.1\% \cdot \sum_{t=1}^{p} Q(D_t)[d]$
RGP [23]	w-event	✓	Uniform	✗	({0.5,1.0}, 1)	({10,50}, u)	MRE $\gamma = u$
RescueDP [27, 28]	w-event	✓	none	✗	([0.1,1], 200)	([40,240], 1)	MAE, MRE $\gamma_d = 0.1\% \cdot \sum_{t=1}^{p} Q(D_t)[d]$
Re-DPoctor [34]	w-day event	✗	none	✗	([0.5,1.5],14)	([7,35],1)	MAE, MRE $\gamma = 0.05\% \cdot \sum_{t=1}^{p} Q(D_t)$
PeGaSuS [7]	event-level	✓	Uniform	✗	({0.01, 0.1},1)	n.a.	MAE, true positive rate
Local DP[c] [13]	local w-event	✓	none	✗	({1.1,1.9},4)	([10,100],1)	MAE, RMSE
STBD [22]	(w,n)-DP	✓	Uniform	✗	([0.2,1.0],120)	([40,200],1)	MAE
DPS [14]	local w-event	✗[d]	Uniform	✗	-	([0.01,1], u)	unspecified 'average error'
AdaPub [30]	w-event	✓	none	✗	([0.1,0.9], 100)	([40,200], 1)	MRE $\gamma_d = 1\% \cdot \sum_{t=1}^{p} Q(D_t)[d]$
DADP [31]	distr. w-event	✓	none	✗	([0.1,1],40)	(1.0, [20,200])	MAE, MRE $\gamma = 0.1\% \cdot \sum_{t=1}^{p} Q(D_t)$
ToPS [29]	event-level	✓	none	✗	([0.01,0.5],1)	n.a.	mean squared error
LPD-IDS [25]	local w-event	✓	both	✗	({0.5, 2.5},20)	({10, 50},1)	MRE $\gamma_d = u$, event monitoring ratio

[a]FAST used for FAST$_w$: http://www.mathcs.emory.edu/~lxiong/aims/FAST/
[b]Only the user-level mechanism is compared to Uniform.
[c]The publication does not propose a name. For convenience, we use the name Local DP relating to the query the mechanism computes.
[d]Privacy proof missing.

3.1 Mechanism Set $\mathbb{M}$

Below, we state five requirements ($\mathbb{M}$-R1) to ($\mathbb{M}$-R5) that the mechanism set $\mathbb{M}$ needs to fulfill in order to provide comparability. We further discuss to which extend previous works address these requirements (summarized in Table 3).

($\mathbb{M}$-R1) Proofing the Desired Privacy Definition. Upon selecting a mechanism, the most fundamental requirement is that the mechanism provides the desired privacy guarantee, i.e., w-event DP in our case. We distinguish two cases : (1) If the definition is used directly, the authors need to prove that the definition is satisfied. (2) If a novel guarantee is proposed, e.g., a generalization of w-event DP, the authors need to prove that their mechanism satisfies the novel guarantee. Further, they need to state how the mechanism can be parameterized such that it fulfills w-event DP. Though this appears to be self-evident, our survey reveals that there are mechanism propositions without a privacy proof (cf. Table 3).

($\mathbb{M}$-R2) Inclusion of Baseline Mechanisms Uniform and Sample. In the original w-event DP publication [20], the authors propose two baseline algorithms: Uniform and Sample. Their design is based on the fact that any mechanism introduces two types of errors into the stream, namely the perturbation and the approximation error. One of them is dominant for each baseline. Specifically, the perturbation error occurs in the Perturbation function that perturbs $Q(D_t)$ by adding noise. It is defined as the difference between the true query result $Q(D_t)$ and the perturbed one p_t. The approximation error occurs when a mechanism does not sample and hence approximates the current query $Q(D_t)$ with the last released sanitized result r_l. It is defined by the difference between the true query result $Q(D_t)$ and the last released one r_l. Especially if the query result fluctuation is small, the approximation error is also small.

Uniform samples every timestamp by allocating $\epsilon_t = \frac{\epsilon}{w}$ budget for perturbation; hence, only a perturbation error is introduced. By contrast, Sample only samples a new query result every w^{th} timestamp and approximates the query results at the remaining timestamps. Thus, it uses the total budget, i.e., $\epsilon_t = \epsilon$, for perturbation and its error is dominated by the approximation error. As a result, we suggest to include *both* baseline mechanisms, as they allow to study the dominant error type and help quantifying the improvement of a newly proposed mechanism. However, our literature study reveals that 7 out of a total of 16 publications do not include any of these baselines. Moreover, 7 publications only compare to one of the baseline mechanisms.

($\mathbb{M}$-R3) Availability of Mechanism Implementations. Most mechanisms proposed in literature are intrinsically complex. For instance, the sampling decision of multiple mechanisms rely on a so-called *proportional-integral-derivative (PID) controller* [3, 16, 21, 28] or Kalman filter [16, 28, 33]. Aiming at validating experimental results, we observed that minor differences in the implementation or parameter initialization can have a significant effect on a mechanism's utility. For example, is the query result in the Filtering function rounded to the query domain or not. Therefore, we advocate to make implementations available online to provide additional insights and facilitate the comparison. Our literature survey reveals that only one out of 16 publications provide access to their implementation. Moreover, we got access to re-implemented sources by contacting the authors of the original w-event publication [20] which highly helped to validate our own implementations.

($\mathbb{M}$-R4) Private Parameter Determination. Generally, all parameters of a mechanism that are computed on the true stream need to be computed in a private way [18], e.g., the number of rows which is private information. None of our surveyed publications address this requirement explicitly, even though not all mechanisms sanitize these parameters. However, verifying this requirement without having access to the concrete mechanism implementation ($\mathbb{M}$-R3) is impossible. Using our benchmark (cf. Section 4), we identify that three out of 10 w-event mechanism do not fulfill this requirement. To solve this issue, we suggest to follow the proposal of [18] to use *mechanism repair functions*.

($\mathbb{M}$-R5) Homogeneity of Background Knowledge. Most mechanisms use components, like PID controllers [3], that have parameters as well. Background knowledge of the domain is required to set them optimally. However, it is important to use them consistently in the benchmark to provide a fair comparison of all mechanisms [18].

3.2 Data Stream Set $\mathbb{S}$

Ideally, an empirical comparison consists of two parts: First, a sequence of micro benchmarks on artificial data is conducted to study the effect of stream properties on a mechanism's utility. Second, a canon of real-world streams is used to reflect use cases.

($\mathbb{S}-R1$) Artificial Streams Reflecting Stream Properties. Our survey reveals that artificial streams are rarely used, e.g., in [25]. Even though related work [15, 20, 30] indicates that a mechanism's performance depends on fluctuations and the sparsity of the stream, further investigations are missing. Therefore, the identification of stream properties that are relevant for either the mechanism's utility or the reflection of real-world data remains an open challenge. To this end, we propose and discuss relevant stream properties when instancing our benchmark (cf. Section 4).

($\mathbb{S}-R2$) Available Real-World Streams with Reproducible Preprocessing. Our literature survey reveals that most approaches focus on real-world streams from specific use cases, e.g., location monitoring. Even though multiple publications use the same streams, the respective study results are not necessarily comparable. The reason is that the streams are preprocessed which highly varies between most studies. We illustrate this fact with two examples referring to the most common stream data, namely WorldCup. Its raw dataset contains the logs of 89,997 URLs of the FIFA 1998 Soccer World Cup website. (a) [20] refers to all 89,997 web pages, i.e., dimensions, while [28] samples 2,000 of them. The reported utilities in both publications indicate that the same mechanisms may have a highly different utility depending on the conducted preprocessing. (b) [32] aimed to reproduce the results of [20]. Only after having access to the preprocessed stream, kindly provided by the original authors, an additional preprocessing step was identified, i.e., normalizing the counts. Both examples suggest that the result deviation originates from stream preprocessing. In many other cases, the reason remains unknown. Since we are aware that due to license issues most publications must not publish their preprocessed streams, it is particularly important that all preprocessing steps are well documented and publicly available [6, 24].

3.3 Privacy Requirements Set $\mathbb{P}$

In the w-event DP framework, data owners express their privacy requirements by a tuple (ϵ, w) where ϵ is the available privacy budget and w is the window length. However, there is no clear consensus in the literature regarding the range of examined privacy budgets, window sizes, and their combination (cf. Table 3). In all publications, the authors conduct two types of experiments:

($\mathbb{P}$ - R1) Vary-ϵ. The authors examine the effect of ϵ for a fixed value of w. Mostly, ϵ is varied between 0.1 and 10.

($\mathbb{P}$ - R2) Vary-w. The authors examine the effect of w for a fixed value of ϵ, mostly $\epsilon = 1$.

However, there is no consensus regarding the window size w for both types of experiments. The w-values even differ for the same stream. The overall tendency is a lower bound of $w > 10$ and an upper bound in the low hundreds.

3.4 Error Metrics Set $\mathbb{E}$

Researchers typically compute an error metric between the true and the sanitized stream to determine the utility of a mechanism. As shown in Table 3, most studies use the mean absolute error (MAE) or the mean relative error (MRE), as defined in Section 2.3. However, there are subtle differences in the error calculation; in particular, in the selection of the sanity bound of MRE. For instance, [28] uses a data-dependent sanity bound γ, whereas [8] fixes $\gamma = 1.0$. In three publications, the sanity bound is not stated, even though the used streams contain query results of 0, requiring $\gamma > 0$. Moreover, since mechanisms rely on random values, two runs of the same mechanism using the same combination of stream and privacy requirements may result in a highly different error. Consequently, as suggested in [18], we suggest to run each combination multiple times, and compare the average and the 0.95-quantile of the error.

4 BENCHMARK DEFINITION

We now introduce a benchmark for w-event DP mechanisms aiming at the comparability of experimental results. The benchmark is defined based on the elements identified in Section 3 and meets all comparability requirements.

Table 4 gives a brief overview of the element selection which results in the so far largest empirical study, comprising 252,000 single experiments, i.e., mechanism runs. Next, we discuss how to meet the requirements of each element and argue how to ensure the validity and comprehensiveness of the study results.

4.1 Mechanism Set $\mathbb{M}$

We first discuss the selection of mechanisms in our benchmark. We give a detailed discussion on meeting all the identified requirements from Section 3 to ensure the comprehensiveness and validity of the intended results.

($\mathbb{M}$-R1)-($\mathbb{M}$-R2) Considered Mechanisms. We include (1) the baseline mechanisms Sample and Uniform, as well as (2) *all* mechanisms found in our literature study that either (a) support w-event DP directly or can be parameterized such that they achieve w-event DP. We exclude mechanisms that provide local or distributed w-event

Table 4: Benchmark instantiation of the 4-Tuple $(\mathbb{M}, \mathbb{S}, \mathbb{P}, \mathbb{E})$.

Elem.	Instantiation
$\mathbb{M}$	(1) Baselines: Sample [20], Uniform [20]; (2) Competitors: FAST$_w$ [20], DSAT$_w$ [21], BD [20], BA [20], RescueDP [27, 28], AdaPub [30], PeGaSuS [7]
$\mathbb{S}$	(1) 20 artificial seasonal streams with dim = 1 (2) 8 real-world streams from Table 5: WorldCup, Taxi Porto, Flu Outpatient, Taxi Beijing, State Flu, Flu Death, Retail, and Unemployment
$\mathbb{P}$	(1) Vary-ϵ: $\epsilon \in [0.1, 1.0]$, $w = 120$ (2) Vary-w: $w \in [40, 200]$, $\epsilon = 1.0$
$\mathbb{E}$	(1) Average MAE over 100 runs (2) Average MRE with $\gamma_d = 0.1\% \cdot \sum_{t=1}^{p} Q(D_t)[d]$ for dimension d over 100 runs (3) Comparison of average error with 0.95 quantile of error

DP, since their utility is lower than the one of pure w-event DP mechanisms by definition [9, 25, 31]. We further include (b) all mechanisms used as competitors for a mechanism in (a). According to Table 3, criterion (a) applies to FAST$_w$ [20], DSAT$_w$ [21], RGP [23], SecWeb [26], RescueDP [27, 28] and AdaPub [30]. Since SecWeb is a prequel of RescueDP, we do not include SecWeb in our benchmark. We furthermore exclude RGP, since it is only applicable to hierarchic location count streams. Criterion (b) includes PeGaSuS [7] being a competitor of AdaPub. We do not include Uniform with backwards smoothing (competitor of PeGasuS) since preliminary experiments revealed that it does not yield a substantial utility improvement compared to Uniform. Since PeGaSuS provides event-level DP only, we adjust it such that it provides w-event DP. Inspired by the Uniform mechanism, we do this by providing a budget of $\epsilon_t = \frac{\epsilon}{w}$ per timestamp t. Analogously to the proof in [20], PeGaSuS then fulfills w-event DP.

($\mathbb{M}$-R4) Private Parameter Determination. A pivotal requirement is that all mechanism determine data-dependent parameters in a private way. As discussed in Section 3, we use mechanism repair functions whenever we find parameters that are not determined in a private way. Specifically, we use the following repair functions.

DSAT$_w$ Repair Function. This mechanism uses the number of rows, i.e., total count at each timestamp, in the stream. Since streams that feature a different number of rows are neighboring, this is private information. We repair DSAT$_w$ as follows: Calculate the total counts at the first timestamp and *perturb* it by spending 10% of the privacy budget allocated for $t = 1$. To keep the privacy guarantee, we reduce the perturbation budget allocated at timestamp $t = 1$ accordingly. If the sanitized total count equals zero, the repair function uses the value 5,000 also used in the original publication [21].

BD/BA - Column Partitioning Repair Function. Mechanisms BA and BD may use an optimization requiring to group the dimensions based on their correlation. Since non-coincidental correlation among dimensions is private information, it needs to be determined in a private way. This also holds despite the observation in the original publication [20] that both mechanisms are very sensitive towards this parameter. The original results indicate the number of groups should be rather small. For instance, on the WorldCup stream, they achieve the best results with 150 groups for 89,997 dimensions [20]. Consequently, we repair BD by using 0.2% of the dimensions as number of groups. We do not group in BA, because initial tests suggest no significant improvement.

($\mathbb{M}$-R5) Homogeneity of Background Knowledge. All mechanisms (except for Uniform, Sample, BD, and BA) use components that rely on configuration parameters, e.g., a PID controller. A mechanism specific parameter is set as given in the publication. If mechanisms share parameters, we set these parameters consistently in all mechanisms. Specifically, there are two parameters used in more than one mechanism: (1) The *desired sampling rate* used in Fast$_w$ and DSAT. The FAST publication [16], a subroutine of FAST$_w$ [20], uses a rather high (15%) sampling rate, while the publication proposing DSAT$_w$ uses a rather small one (1%). As preliminary experiments revealed, both mechanisms tend to provide higher utility for higher rates. Therefore, we use 15% as desired sampling rate in both mechanisms. (2) The parameters of the PID controller that is used in FAST$_w$, RescueDP, and DSAT$_w$. While the publications proposing RescueDP [27, 28] and FAST [16] suggest the same parameter values, the values used in DSAT$_w$ [21] differs. However, while in DSAT$_w$, the PID controller controls the change in the sampling rate, in RescueDP and FAST$_w$, it controls the change in the sanitized values. Consequently, the operational purpose of the use of the PID controller is different. As a result, we use the parameters suggested in the respective publication.

($\mathbb{M}$-R3) Mechanism Implementation. A correct implementation of the mechanisms is a key factor to ensure result validity. We ensure validity of our results by the following four key principles: (a) favor original implementation, (b) re-use of well-known mechanism parts, (c) consistency checks of independent implementations, and (d) contact original authors if necessary. Next, we explain these principles in more detail: First, in case the publication proposing a mechanism offers an implementation, we use this implementation. However, as shown in Table 3, this only holds for one mechanism, namely, FAST$_w$. Second, multiple mechanisms use the same component (e.g., the sampler), which is itself available open source. For instance, FAST$_w$ uses a Kalman filter and PID controller. In such cases, we use this component consistently in all mechanisms. Third, all mechanisms are implemented redundantly and independently by up to three different people, possibly all leading to consistent results. Finally, the results are consistent among our implementations but highly deviate from the results in the original publication. Consequently, we contacted the original authors of w-event DP [20] and thankfully received implementations from them. This not only helped to ensure that all baselines and advanced mechanisms proposed in [20] are correct, but also for principle (b) since we had more mechanism parts for re-usage.

4.2 Data Streams Set $\mathbb{S}$

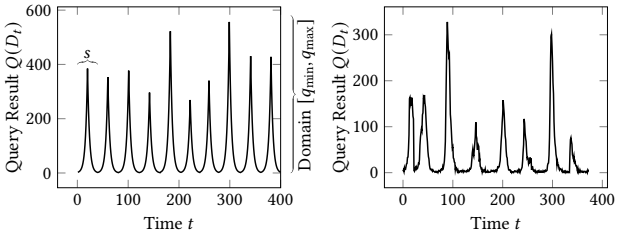
Concerning data streams, we meet the requirements from Section 3 as follows: First, we conduct a series of micro benchmarks with artificial data (i.e., $\mathbb{S}$-R1). Second, we conduct experiments on a comprehensive set of data streams used in literature (i.e., $\mathbb{S}$-R2).

Table 5: Requirement ($\mathbb{S}$ – *R2*): Availability of real-world streams used in prior work: ✓yes, ✗no/removed, ✓partially.

Stream	Publicly available	Used in reference	Limitations
WorldCup	✓	[5, 8, 20, 26, 28?]	raw data only
Rome traffic 1	✗	[20]	-
Montreal traffic	✗	[5]	-
Rome traffic 2	✗	[23]	-
Heart rates	✗	[34]	-
Taxi Porto	✓	[21, 22, 28, 31?]	raw data only
San Joaquin / Oldenburg	✓	[21, 28?]	data generator only
WiFi traces 1	✗	[7]	-
WiFi traces 2	✗	[13]	-
GeoLife	✗	[22]	-
Flu Outpatient	✓	[15]	only other years and ages
traffic Seattle	✗	[15]	-
Unemployment	✓	[15]	-
US census	✓	[21]	raw data only
TDrive	✓	[21, 25]	-
APASCologne	✓	[14]	-
State Flu	✓	[30]	-
Flu Death	✓	[30]	only from other seasons
Retail	✓	[30]	-
Nice ride	✓	[31]	-
DNS	✗	[29]	-
Fare	✓	[29]	raw data only
Kosarak	✓	[29]	raw data only
POS	✗	[29]	-
Foursquare	✓	[25]	-
Taobao	✗	[25]	requires account

($\mathbb{S}$-R1) Artificial Streams Reflecting Stream Properties. The intention behind using artificial data is to study the influence of relevant stream properties on a mechanism's utility in a structured way. Generating meaningful artificial data is challenging. For streams in general, there are various properties known to have an influence on data processing, e.g., dimensionality, seasonality, level, and trend [19]. However, neither the properties nor their influence on the utility of a mechanism on real-world streams used in previous studies have been investigated so far. Next, we (a) analyze which of these properties do occur in the real-world streams listed in Table 5, and (b) describe the design of our artificial data generator that allows to investigate each of the properties in isolation. Finally, we present the data generator itself.

Dimensionality. The streams in Table 5 provide a dimensionality between 1 and 80,000. In our micro benchmark, however, we consider univariate query results per timestamp, i.e., $dim = 1$. We aim to understand a mechanism's ability to retain utility of the stream using clever budget allocation, sampling, filtering, and leveraging the inertia of the stream. We intentionally exclude the additional utility improvement of some mechanisms, gained by taking advantage of correlated dimensions, in our micro benchmarks by setting $dim = 1$. The reason is that introducing known correlations in multi-dimensional streams is highly challenging.



(a) First 400 time stamps of one artifical stream. **(b) Flu Death stream.**

Figure 1: Artificial data stream with domain $[0, 600]$ and expected season length $s = 40$ vs. a real-world 1D stream.

Level. Most seasonal streams feature inter-seasonal downtimes, i.e., $Q(D_t)$ is close to zero (cf. Figure 1b). The minimum query result is usually also the most frequent one. To decouple the level from the seasonality, we quantify the level by the minimum query result $q_{\min}$. The minimum query result of the stream influences a mechanism's utility in case the filtering technique *truncating* (cf. Section 2.2) is applied. For queries like Count and Histogram, truncating filtering rounds the negative perturbed query results to zero. Consequently, whenever the Laplace mechanism adds a negative amount of noise (e.g., -10) which holds in half of the cases, the mechanism releases the true query result instead. By contrast, the mechanism introduces a relative error of 100% if $Q(D_t) = 10$. Hence, truncating reduces the noise by taking advantage of the query domain, especially at low levels of the stream. Consequently, we do not truncate the sanitized query result in our micro benchmarks. That way, the utility for streams of different levels is equal if the other properties are equal and we do not need to investigate the mechanism utility for varying stream levels.

Seasonality. We observed that most real-world streams have a seasonality, with an exponential growth and shrinking phase. The maximum query result $q_{\max}$ highly varies from stream to stream. The perturbation and approximation error, however, are clearly influenced by the length of the seasons s and the amplitude $a = q_{\max} - q_{\min}$ where $q_{\min}$ is the minimum query result. Thus, we test the mechanism utility with respect to both. Note that since $q_{\min} = 0$ the following holds: $a = q_{\max}$. In our micro benchmarks, we generate streams for every combination of $s \in \{40; 60; 80; 10; 120\}$ and $a = q_{\max} \in \{10; 100; 1{,}000; 10{,}000\}$ reflecting values observed in real-world streams.

Trend. We do not observe a trend in the sanitized streams listed in Table 5. Therefore, we do not consider this property.

Data Generator. Figure 1a shows example data we generated using our data generation algorithm (cf. Algorithm 2). Generally, the artificial data shall be similar to one-dimensional streams used in other studies. For the depicted data, we use $p = 400$ timestamps, amplitude $a = q_{\max} = 600$, and an average season length $s = 40$. Not all periods have exactly the same length, we therefore dice the length of each season with $\mathcal{G}(s = 40, 2)$. For the growing phase, we use an exponential growth function $Q(D_t) = e \cdot Q(D_{t-1})$ with $e = 1.5$. The shrinking phase is symmetric to the growing phase.

Next, we also mimic inter-seasonal downtime by dicing the season minimum with with $\mathcal{G}(s = 8, 2)$, i.e, some value close to zero. Since the maximum value of the stream generated this way depends on the actual length of the season and the diced minimal values, we need to normalize the maximum value with the desired amplitude a. Finally, the stream might be too long because the algorithm generates the stream season-wise. Thus, we return the stream prefix until timestamp p.

Algorithm 2 Data Generator

```
1:  function generateStream(p, s, a)
2:      t ← 1, e ← 1.5
3:      while t < p do                          ▷ Each loop generates one season
4:          sl ← G(s, 2)                        ▷ Dice season length
5:          val ← G(8, 2)                       ▷ Dice season minimum, close to 0
6:          Q(D_t) ← val; t++
7:          for i = 1 to sl/2 do
8:              val ← e · Q(D_{t−1})            ▷ Exponential growth
9:              Q(D_t) ← val; t++
10:         end for
11:         ...                                 ▷ Symmetric shrinking phase
12:     end while
13:     max ← max{Q(D_1), .., Q(D_{p−1})}
14:     for i = 1 to p do
15:         Q(D_i) ← Q(D_i)/max · a             ▷ Ensure desired amplitude
16:     end for
17:     return Q(D_1), .., Q(D_p)               ▷ Ensure correct length
18: end function
```

(𝕊-R2) Publicly Available Real-World Streams with Reproducible Preprocessing. For comprehensiveness, we use all real-world streams that are freely available and at least used once to evaluate a w-event DP mechanism. According to Table 5, the following streams qualify: WorldCup, Taxi Porto, Flu Outpatient, TDrive, State Flu, Flu Death, Retail and Unemployment. All of them use a query Q with $\Delta Q = 1$. As far as useful and possible, we preprocess them according to one of the respective publications. To facilitate comparability and reproducibility, all preprocessing steps are available at our project website[3].

4.3 Privacy Requirements Set $\mathbb{P}$

Inspired by most of the experimental studies found in the related work, we also conduct the vary-ϵ and vary-w experiments, fulfilling ($\mathbb{P}$-R1) and ($\mathbb{P}$-R2). For the vary-ϵ experiment, we select a reasonably large value for parameter $w = 120$ and vary $\epsilon \in [0.1, 1]$ with an increment of 0.2. For the vary-w experiments, we use $\epsilon = 1$ as most studies do. Furthermore, we vary $w \in [40, 200]$ with a w increment of 40, such that there is an overlap with various other studies.

4.4 Error Metrics $\mathbb{E}$

Since the mechanisms rely on randomness, the utility can differ highly for the same combination of privacy requirements and stream. Following various studies from the related work, we run each experiment 100 times and use the average MAE and average MRE (with $\gamma_d = 0.1\% \cdot \sum_{t=1}^{p} Q(D_t)[d]$ for dimension d) to quantify the error the mechanisms introduce into the sanitized data. Besides the average error, we quantify the variance of the error as suggested by [18] for static (i.e., standard) DP. To this end, we measure the 0.95 quantile of MAE and MRE reflecting a 'risk averse' data owner.

[3] https://dbresearch.uni-salzburg.at/projects/dpbench/index.html

5 EXPERIMENTAL RESULTS

We perform an experimental study by executing our benchmark as instantiated in Table 4. The goal of this study is to gain new insights into the strengths and weaknesses of existing mechanisms. Further, we analyze the influence of stream properties on a mechanism's utility using our artificially generated streams and verify whether the results also hold for real-world streams.

5.1 Artificial Streams

With the artificial streams, we aim at understanding the effect of the two identified stream properties seasonal period length s and amplitude a on a mechanism's utility. Specifically, we are interested in the perspective of a data administrator aiming at selecting a mechanism for a given stream and privacy requirement. Consequently, we formulate the following two research questions:

- **(RQ1)** Are stream properties decisive for mechanism selection?
- **(RQ2)** If so, can we recommend a mechanism and/or function design for a given seasonal period length s, an amplitude a, and privacy requirements (ϵ, w)?

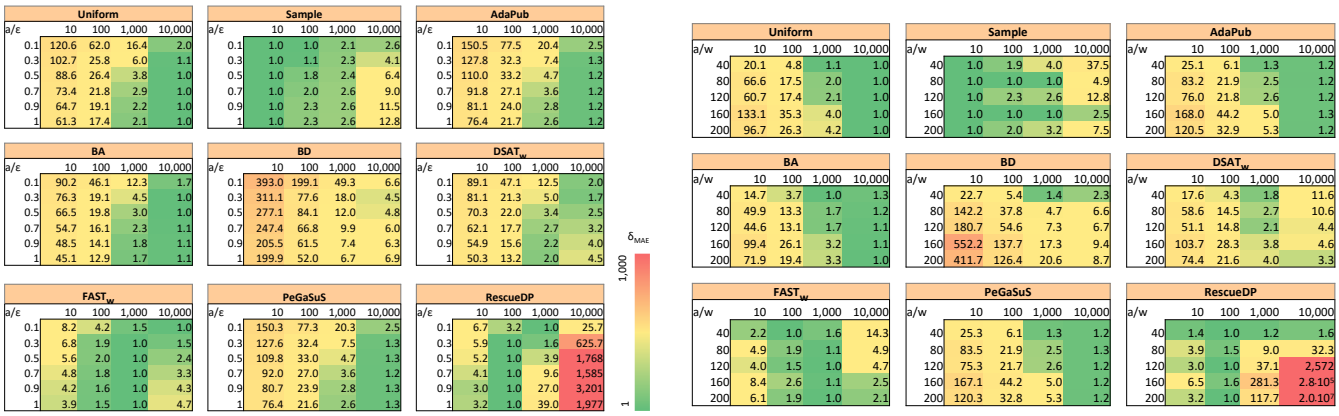
For brevity, we subsequently focus on the mean MAE, short MAE, to answer these questions. This is valid because the result patterns for the 0.95 quantile of MAE and MRE are similar. To make the mechanism's MAEs comparable over all streams and privacy requirements, we consider the MAE deterioration $\delta_{\text{MAE}}(c)$ for a specific combination of mechanisms, stream properties, and privacy requirements $c = (m \in \mathbb{M}, (s, a), (\epsilon, w) \in \mathbb{P})$. For a specific combination c, the MAE deterioration compares the MAE of mechanism m to the best mechanism m' with minimum MAE:

$$\delta_{\text{MAE}}(m, (s, a), (\epsilon, w)) = \frac{\text{MAE}(m, s, a, \epsilon, w)}{\min\{\text{MAE}(m', s, a, \epsilon, w) \mid m' \in M\}}$$

We present the MAE deterioration on artificial streams in Figure 2. The color gradient marks small values in green, i.e., good utility, and large values in red, i.e., bad utility. Subsequently, we discuss the results with respect to the research questions.

5.1.1 (RQ1) Are stream properties decisive for mechanism selection? For answering this question, we investigate how the stream properties amplitude a and period length s as well as privacy requirements ϵ and w influence MAE. The raw MAE results of the vary-ϵ and vary-w experiments (not illustrated) indicate that the utility behaves as expected for most mechanisms. Specifically, they show a proportional MAE increase or decrease towards a change of the privacy requirements. For instance, we observe that the MAE declines by roughly a factor of 2 when doubling the available budget ϵ for constant a, s, and w. The only notable exception is RescueDP which hardly benefits from higher budgets when the amplitude $a > 1,000$. This is due to the fact that RescueDP is specifically designed for publishing multi-dimensional data with small amplitudes.

Next, we observe that the period length s is not decisive since the MAE deterioration is equivalent for each s when a, ϵ, and w



(a) Vary-ϵ experiments with $w = 120$. **(b) Vary-w experiments with $\epsilon = 1.0$.**

Figure 2: Heat map of the δ_{MAE} results from the vary-ϵ and vary-w experiments for a period length of $s = 80$.

are fixed. By contrast, the amplitude a is highly decisive for a fixed period length $s = 80$ (cf. Figure 2). For instance, Sample provides the lowest MAE for $a = 10$ and $\epsilon = 0.1$ while AdaPub is the winner for $a = 10,000$ and $\epsilon = 1.0$. Summarizing, mechanisms provide a high utility either for small or for large amplitudes independent of other parameters such as s, ϵ, or w.

5.1.2 (RQ2) Can we recommend a mechanism for specific stream properties and privacy requirements? Considering Figure 2, either Sample or Uniform is among the mechanisms with the smallest MAE for almost every combination of parameters. This is surprising since baseline mechanisms frequently outperform sophisticated mechanisms. We investigate this result by analyzing the parameter settings in which either Uniform or Sample provides the smallest MAE and outline issues regarding hypersensitive data-adaptive sampling of sophisticated mechanisms. We further investigate whether the baseline supremacy also holds for real-world streams.

Uniform supremacy. Our results suggest that mechanism Uniform is among the best for large amplitudes $a \geq 1,000$ and non-restricting privacy requirements, i.e., large ϵ, small w. In general, the relevance of restrictiveness decreases for increasing a. The expected MAE = $\frac{w}{\epsilon}$ of Uniform is data-independent. Thus, we gain little insights on MAE of Uniform in case Uniform is among the best mechanisms in terms of δ_{MAE}. Instead, we identify that the invested budget (e.g., for data-adaptive sampling) of sophisticated mechanisms does not pay off since MAE might exceed $\frac{w}{\epsilon}$. Moreover, we expected that AdaPub and PeGaSuS consistently have a lower error than Uniform when Uniform is among the best. The reason for this assumption is that they differ from Uniform only in an additional Filtering-function, i.e., smoothing the perturbation noise. However, our results do not confirm this expectation since their filtering requires a fraction of the privacy budget ϵ. This investment only pays off in downtimes between the seasons where the query results are fairly stable. Within growing or shrinking phases of a season, the groups usually contain a single timestamp and the mechanism has less budget for perturbation.

Sample supremacy. Comparing Uniform with Sample reveals that Sample's MAE is smaller than Uniforms' if $q_{\max}$ is small and the privacy requirements are restrictive. While Uniform's MAE is data-independent, Sample is guaranteed to be w-independent. Hence, it only depends on the minimum and maximum query results $[q_{\min}, q_{\max}]$ and ϵ. In our case, the minimum value is $q_{\min} = 0$. Thus, the maximum approximation error converges towards $q_{\max}$. This worst case occurs if $Q(D_t) = 0$ for all sampled timestamps and $Q(D_t) = a = q_{\max}$ otherwise. Moreover, the perturbation error is $\frac{1}{\epsilon}$. Thus, the MAE bound is $q_{\max} + \frac{1}{\epsilon}$ and hence independent of w. For instance, Sample's bound with $a = 10$, $w = 100$, and $\epsilon = 1$ is 11 , whereas Uniform's bound is 100, i.e., 10 times larger. Our empirical results reveal that we rarely observe Sample's bound and the observed MAE is several factors smaller. The rational is that Sample has a tendency to publish the most frequent query results very accurately.

Hypersensitive data-adaptive sampling. The small MAEs of Sample for small amplitudes and restrictive privacy requirements suggest that the perturbation error needs to be minimized via sampling. The mechanisms BD, BA, DSAT$_w$, FAST$_w$, and RescueDP feature data-adaptive sampling. The idea is compelling: Instead of hoping that the last release approximates the next timestamps well, the mechanism invests a fraction of the budget ϵ to monitor the stream. In case the mechanism monitors a large enough change, a new query result is releases. However, our results suggest that data-adaptive sampling does not consistently outperform sampling with data-independent rates (as conducted by Sample). Instead, they are only better than Sample when Uniform is better as well. The rational is that data-adaptive sampling features a hyper-sensitivity for small changes in the query result. Specifically, we observe the following tendencies: In case the growing phase of a new season starts, the initial small changes of the query result is well reflected. In addition, the sample timestamp is close to the peak of the first season. Thereby, a large fraction of the available budget is already spent in the growing phase. Thus, data-adaptive sampling is more reluctant in spending budget in the shrinking phase, i.e., large query

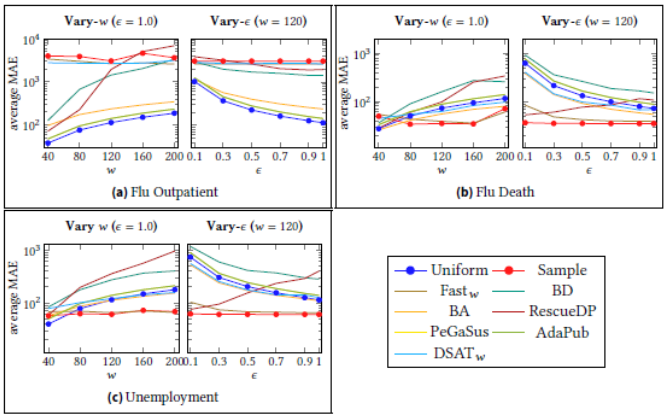


Figure 3: Average error of w-event mechanisms in vary-w and vary-ϵ experiments for one-dimensional real-world streams. Baselines marked with ∘.

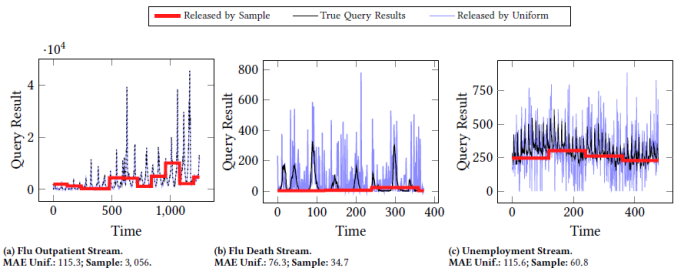


Figure 4: True query result stream of one-dimensional streams as well as the streams released by the baselines Uniform and Sample for $\epsilon = 1.0$ and $w = 120$.

results are produced in the shrinking phase, incurring a high MAE. That becomes worse whenever multiple seasons fit into one window of size w. This holds for all common window sizes and streams.

5.2 One-dimensional Real-World Streams

Next, we evaluate the results on one-dimensional real-world streams. Thereby, we have two objectives: First, we are interested whether the results of real-world streams are consistent with the results on artificial streams, particularly the observed baseline supremacy. Second, we aim at understanding the abstract error measures (e.g., MAE) in context of the data streams. In a nutshell, our key results are: the results on real-world streams are consistent with the micro benchmarks, and common error metrics are not well-suited in the streaming setting.

5.2.1 Confirmation of micro benchmark results. In Figure 3, we depict MAE for all mechanisms and one-dimensional real-world streams. For better visualization, MAE of the baseline mechanisms Uniform (blue curve) and Sample (red curve) are marked with ∘.

Summarizing, the results of the real-world streams confirm the observations in the micro benchmarks. Specifically, we analyze two medium-amplitude streams with $a < 1,000$ (i.e., Flu Death and Unemployment) and one large-amplitude stream (i.e., Flu Outpatient) with a common season maximum of about $2 \cdot 10^4$ (cf. Figure 4). As expected, Uniform has the best MAE for the large-amplitude stream. Notably, MAE is significantly smaller than the expected MAE = $\frac{w}{\epsilon}$, e.g., MAE is almost half as large as expected on the Unemployment stream due to the large amount of timestamps where $Q(D_t)$ is close to 0. The reason is the truncation of the perturbed query result: In many timestamps where Uniform adds a negative noise, a count of 0 is published. Interestingly, we observe a slight utility improvement by PeGaSuS towards Uniform for the Unemployment stream. Sample usually provides the best MAE on the medium-amplitude streams. Only for non-restrictive privacy requirements, i.e., $\epsilon = 1$ and $w = 40$, Uniform and most other mechanism have slightly better MAE. As in the micro benchmarks, data-adaptive sampling is not superior to equidistant data-independent sampling. As in the micro benchmarks, we observe an anomalous behavior of RescueDP: Increasing the available budget does not improve the utility for large-amplitude streams; instead, it has the opposite effect.

5.2.2 Semantics of the Abstract Utility Metric Values. So far, most studies use MAE and MRE metrics to determine a mechanism's utility (cf. Table 3). Considering our results on MAE and MRE, we observe intrinsic anomalies. For example, the utility of Sample appears to be almost independent of the privacy requirements. Thus, we examine the semantics of the abstract error values. To do so, we consider common applications performed on data streams, e.g., forecasting or change detection algorithms. For such applications, the preservation of the stream properties from Section 4.2 is highly relevant. However, there is little knowledge about their relation to MAE and MRE. To this end, we examine the sanitized query results of Sample and Uniform with respect to seasonality (i.e., period length and amplitude) and level. Therefore, we ensure that there is at least one mechanism having a good MAE for every stream due to the baseline supremacy. Our explanations hold in general and are based on exemplary sanitized releases and the real-world streams shown in Figure 4.

Maintaining Seasonal Growing and Shrinking of the Stream. As revealed by the exemplary results in Figure 4, Sample entirely loses its seasonality, independent of the observed MAE. This also holds for streams where Sample performs best. In case the mechanism does not sample multiple times per season, an entire season is approximated with a single value for every timestamp. Thus, small MAE values of Sample suggest that the stream contains a large amount of similar query results which the mechanism likely hits upon data-independent sampling. Simply releasing the sanitized query result at the first timestamp for *every* subsequent timestamp yields a similar utility for all streams used in studies so far.

Uniform maintains the seasonality well when amplitudes are large compared to the noise introduced for sanitation. As the expected noise is $\frac{w}{\epsilon}$, a data administrator only needs to know the amplitude to decide whether Uniform delivers acceptable utility. However, this is not reflected by MAE (nor MRE). For instance, MAE in Figure 4b is smaller than in Figure 4a, despite the seasonality can be clearly observed in (a) but not in (b). Hence, MAE has no meaning for maintaining seasonality. However, there is a relation between MAE and level maintained by Uniform, as we discuss next.

Table 6: Properties of multi-dimensional streams. The query result distribution is the distribution of the true query results over all dimensions of the preprocessed streams, including optional dimension sampling.

Stream S	dim	Length p	Query result distribution		
			$q_{\min}$	$q_{\max}$	90% quantile
StateFlu	51	492	0	11,452	924
TDrive	100	672	0	39,871	1,772
Retail	1,298	374	0	372,306	15,089
TaxiPorto	1,298	672	0	317	2
WorldCup	1,298	1,320	0	16,928	0

Maintaining Level and Amplitude of the Stream. Recap that the true level of the stream is defined by $q_{\min}$ and the true amplitude $a = q_{\max} - q_{\min}$. The level and amplitude of the sanitized stream released by Uniform depend on the true level and amplitude. In case $q_{\min} > \frac{w}{\epsilon}$, i.e., the level is higher than the expected noise, the sanitized stream released by Uniform features the domain $[q_{\min} - \frac{w}{\epsilon}, q_{\max} + \frac{w}{\epsilon}]$. This can be observed in Figure 1c where the measured MAE of 115.6 fairly equals the expected MAE of $\frac{w=120}{\epsilon=1} = 120$. By contrast, $q_{\min}$ is close to 0 in Figure 1b, i.e., the minimum possible value of a count query. Thus, truncating count queries lead to an expected level change of $[\max(q_{\min} - \frac{w}{\epsilon}, 0), q_{\max} + \frac{w}{\epsilon}]$.

Since Sample has a low perturbation error, it does not enlarge the domain , i.e., Sample only publishes values within the original minimum and maximum values. However, the sanitized streams usually miss the seasonal peaks of the true streams. Large MAE values, specifically values exceeding Uniform's MAE, indicate that the stream contains large amplitudes which is poorly reflected in Sample's released stream. Small MAE values, in turn, indicate that there are no large seasonal changes and Sample approximates small counts very accurately.

5.3 Multi-dimensional Real-World Streams

We now present the results (cf. Figure 5) of the multi-dimensional streams in Table 6. We aim to confirm the results obtained on one-dimensional streams. Moreover, we analyze adaptive dimension grouping to improve the utility for multi-dimensional streams.

The key idea of *adaptive dimension grouping* is to find a group g of dimensions that have a similar query result, i.e., that are correlated. This can be exploited in two ways: First, the sampling decision is performed per group in BD. Then, groups with frequently changing query results are sampled more frequently than groups with stable query results. Second, the grouping is exploited in the perturbation function for AdaPub and RescueDP. Specifically, the mechanism perturbs the sum of the query results over all dimensions in group g and then assigns each dimension the average of the perturbed sum. This reduces the expected perturbation error from $\frac{1}{\epsilon_t}$ to $\frac{1}{\epsilon_t \cdot |g|}$ [27]. Hence, with increasing dimensionality the perturbation error is highly reduced.

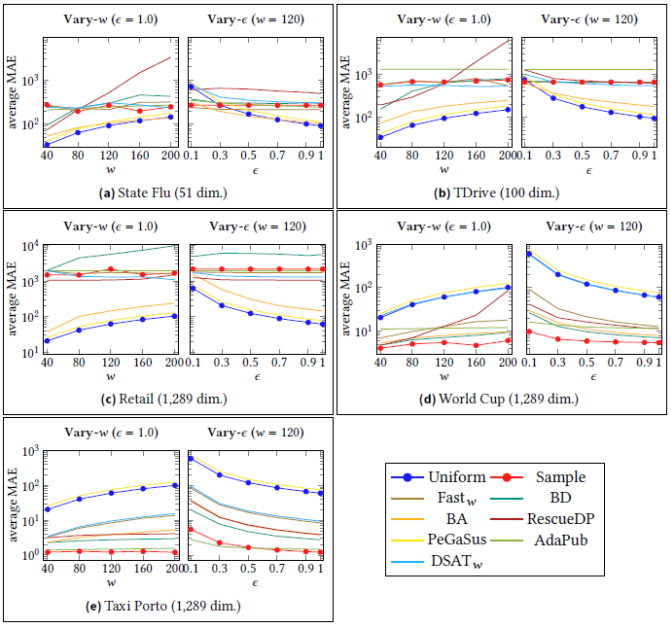


Figure 5: Average error for vary-w and vary-ϵ experiments on multi-dimensional streams. Baselines marked with ∘.

Recap that we observe a baseline supremacy for one-dimensional streams. The amplitude and privacy requirements are decisive factors between Uniform and Sample as well as hypersensitive data-adaptive sampling. Generally, Figure 5 and Table 6 confirm both observations for multi-dimensional streams.

As expected, Uniform is among the best mechanisms for StateFlu, TDrive, and Retail with amplitudes $> 10,000$. Only for small ϵ-values on stream StateFlu, a couple of other mechanism outperform Uniform. Sample is among the best mechanisms on the WorldCup and TaxiPorto stream. However, AdaPub also provides low errors and outperforms Sample for certain ϵ-values. This is interesting since AdaPub has low errors for one-dimensional streams iff Uniform is among the best mechanisms. This suggests that WorldCup and TaxiPorto significantly differ from the other streams. Table 6 reveals that both streams are sparse. Specifically, the query result is very small or even zero for most timestamps and dimensions.

We further observe nearly constant errors for AdaPub in seven experiments and for RescueDP in one experiment. The rationale is that the number of groups converges to one over time, i.e., the mechanism releases the same query result for all dimensions. The perturbation error is low if all dimensions are in one group, i.e., the mean error is only slightly influenced by w and ϵ.

The mean error of BD for these two streams is remarkable: In the micro benchmark and on one-dimensional streams, BD is never among the best mechanisms. However, BD is among the best mechanisms for WorldCup and TaxiPorto. Unfortunately, our results do not show whether this phenomenon is related to grouping.

6 TAKEAWAYS

The primary outcome of our experimental study are takeaways that are relevant for practitioners as well as researchers.

6.1 Takeaways for Practitioners

Our takeaway for practitioners forms a catalog of three recommendations that aim at understanding and controlling the expected utility of w-event DP mechanisms. It targets at data owners and administrators who are not experts in differently privacy but have a sophisticated background knowledge in data analysis.

Data Owners: Meaningful Window Size. The data owner is responsible for selecting the privacy requirements ϵ and w. By definition of w-event differential privacy, the window size w refers to the length of the longest event-sequence the mechanism aims to protect with privacy budget ϵ. The selection of w is clearly use-case dependent. However, our literature study suggests that there is a tendency for investigating unnaturally large values of w which causes large perturbation noise. For seasonal data, the length of a season s may serve as a natural upper bound for w. Our recommendation for data owners is to specify the event-sequence as the maximum length of trajectory in the location monitoring use case.

Data Administrators: Meaningful Utility Metrics. Data administrators are responsible for selecting a mechanism. Our results suggest that abstract error metrics (e.g., MAE or MRE) hardly allow conclusions on whether a mechanism is able to conduct frequent analysis tasks on streams, e.g., forecasting or anomaly detection. Thus, we recommend to select mechanisms that provide high utility with respect to an application specific metric or to investigate the semantics of abstract errors (e.g., MAE) w.r.t. the application.

Data Administrators: Consider the Selection of Baselines. Our study indicates that the Uniform or Uniform-Sample hybrid mechanism is competitive with regards to data utility and expected error. Specifically, we recommend to use Uniform if an expected error of $\frac{w}{\epsilon}$ is sufficient and query results are required for each timestamp, e.g., because one targets at *instant* change detection. If an instant change detection is not needed, time can be traded to minimize the perturbation error using a Uniform-Sample hybrid mechanism. This mechanism samples every x^{th} timestamp; thus, releasing more accurate query results at sampling timestamps than Uniform, i.e., the perturbation error at sampling timestamps is reduced from $\frac{w}{\epsilon}$ to $\frac{w}{\epsilon \cdot k}$. In combination with selecting a meaningful value for w, this mechanism may provide sufficient utility for many applications.

6.2 Takeaways for Researchers

Our takeaway for researcher primarily targets the function design of the w-event mechanism framework. We discuss the functions according to their order in Algorithm 1.

inSamplingPoint-Function. Currently, in case the mechanism does not sample, the current query result is approximated with the last sanitized query result. This works well for timestamps between the seasons when the counts remain stable. However, it yields high errors in a growing or shrinking phase of a stream. Consequently, we propose to investigate mechanisms that consider the seasonal nature of streams upon approximation. For example, mechanisms could invest time and budget to learn a model of the stream (e.g., using machine learning in a differential private way) when starting to release a new stream,. The model can be used for sampling decisions as well as predictions on whether the stream is currently in a growing or shrinking phase. If the change in the stream is not large enough to provoke sampling, the mechanism can correctly approximate based on the latest trend. Note that this is orthogonal to filtering based on time-grouping since the filter is only applied at sampled timestamps.

BudgetAllocation-Function. We observe that mechanisms allocate budget optimistically, trying to accurately reflect small changes in the stream, e.g., mechanism BD allocates half of its remaining budget per sampled timestamp. However, our results indicate that this yields low utility when the stream contains large amplitudes. Homogeneously distributing the budget over sampling timestamps usually provides in the best utility. Thus, mechanisms may limit the number of sampling timestamps in the current window.

Perturbation-Function. Our recommendation regarding perturbation refers to mechanisms using dimension grouping. We frequently observe that the dimensions gather into few or even a single group and hence uncorrelated dimensions are grouped together. We recommend to compute the grouping not only on sanitized query results and consider techniques to ungroup uncorrelated dimensions. Additionally, we propose to question whether researchers should focus on dimension-grouping in future work. The rationale is that dimension-grouping violates privacy in case that the correlation of the dimension query results is spurious. Otherwise, correlated dimensions result from an event that the data owner intends to hide. This may effect multiple rows in a database D_t and not only a single one as presumed in the original definition of differential privacy [11]. The extension of differential privacy with group-differential privacy [12] which states that the increase of ΔQ entirely nullifies the benefit of dimension grouping.

Filtering-Function. Our results suggests that grouping over timestamps with a grouping function that requires budget does not yield a utility improvement. Consequently, we suggest to conduct research on filtering functions that do not require budget.

Finally, in case a researcher proposes a novel mechanism, we strongly recommend conducting an empirical evaluation based on the principles introduced in Section 3. We argue that this is the only way to ensure the comparability of future studies. Most urgently, we recommend to include *both* baseline mechanisms.

7 CONCLUSIONS

We addressed the challenge of comparable empirical studies on w-event differential privacy mechanisms for streams. Based on a comprehensive literature study, we identified common elements of existing studies and formulated requirements for each element to ensure comparability. We introduced a benchmark that meets all requirements and allows for comparable studies. Using our benchmark, we performed the largest empirical study on w-event differential privacy mechanisms so far. Our study revealed valuable insights on existing mechanisms, e.g., a baseline supremacy. Further, we gave advise on mechanism selection and presented promising

research directions in that field. In future work, we aim at a micro benchmark for multi-dimensional query streams. Additionally, we investigate on queries having sensitivity $\Delta Q > 1$, e.g., Sum queries. We hypothesize that the different functions from the w-event mechanism framework are affected heterogeneously by the sensitivity.

REFERENCES

[1] Ergute Bao, Yin Yang, Xiaokui Xiao, and Bolin Ding. 2021. CGM: an enhanced mechanism for streaming data collection with local differential privacy. *Proceedings of the VLDB Endowment (PVLDB)* 14, 11 (2021), 2258–2270.

[2] Mesut E Baran and Arthur W Kelley. 1994. State estimation for real-time monitoring of distribution systems. *IEEE Transactions on Power systems* 9, 3 (1994), 1601–1609.

[3] Richard E Bellman. 2015. Adaptive control processes. In *Adaptive Control Processes*. Princeton university press.

[4] Yang Cao and Masatoshi Yoshikawa. 2015. Differentially private real-time data release over infinite trajectory streams. In *Proceedings of the 16th IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 68–73.

[5] Rui Chen, Yilin Shen, and Hongxia Jin. 2015. Private analysis of infinite data streams via retroactive grouping. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM)*. ACM, 1061–1070.

[6] Xiaoli Chen, Sünje Dallmeier-Tiessen, Robin Dasler, Sebastian Feger, Pamfilos Fokianos, Jose Benito Gonzalez, Harri Hirvonsalo, Dinos Kousidis, Artemis Lavasa, Salvatore Mele, et al. 2019. Open is not enough. *Nature Physics* 15, 2 (2019), 113–119.

[7] Yan Chen, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. 2017. Pegasus: Data-adaptive differentially private stream processing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 1375–1388.

[8] Mian Cheng, Yipin Sun, Baokang Zhao, and Jinshu Su. 2016. An event grouping approach for infinite stream with differential privacy. In *Proceedings of the 10th Asia-Pacific Services Computing Conference (APSCC)*. Springer, 106–116.

[9] Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. 2018. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 ACM SIGMOD International Conference on Management of Data*. ACM, 1655–1658.

[10] Teddy Cunningham, Graham Cormode, Hakan Ferhatosmanoglu, and Divesh Srivastava. 2021. Real-world trajectory sharing with local differential privacy. *Proceedings of the VLDB Endowment (PVLDB)* 14, 11 (2021), 2283—-2295.

[11] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation (TAMC)*. Springer, 1–19.

[12] Cynthia Dwork, Aaron Roth, et al. 2014. The Algorithmic Foundations of Differential Privacy. *Foundation and Trends ® in Theoretical Computer Science* 9, 3-4 (2014), 211–407.

[13] Fatima Zahra Errounda and Yan Liu. 2018. Continuous location statistics sharing algorithm with local differential privacy. In *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 5147–5152.

[14] Soheila Ghane Ezabadi, Alireza Jolfaei, Lars Kulik, and Ramamohanarao Kotagiri. 2019. Differentially private streaming to untrusted edge servers in intelligent transportation system. In *2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*. IEEE, 781–786.

[15] Liyue Fan and Li Xiong. 2014. An Adaptive Approach to Real-Time Aggregate Monitoring With Differential Privacy. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 26, 9 (2014), 2094–2106.

[16] Liyue Fan, Li Xiong, and Vaidy Sunderam. 2013. FAST: differentially private real-time aggregate monitor with filtering and adaptive sampling. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. ACM, 1065–1068.

[17] Nicola J Ferrier, Simon Rowe, and Andrew Blake. 1994. Real-time traffic monitoring.. In *Proceedings of the 2nd IEEE Winter Applications and Computer Vision Workshops (WACVW)*. IEEE, 81–88.

[18] Michael Hay, Ashwin Machanavajjhala, Gerome Miklau, Yan Chen, and Dan Zhang. 2016. Principled evaluation of differentially private algorithms using dpbench. In *Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data*. 139–154.

[19] Rob J Hyndman and George Athanasopoulos. 2018. *Forecasting: principles and practice*. OTexts.

[20] Georgios Kellaris, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias. 2014. Differentially private event sequences over infinite streams. *Proceedings of the VLDB Endowment (PVLDB)* 7, 12, 1155–1166.

[21] Haoran Li, Li Xiong, Xiaoqian Jiang, and Jinfei Liu. 2015. Differentially private histogram publication for dynamic datasets: an adaptive sampling approach. In *Proceedings of the 24th ACM international Conference on Information and Knowledge Management (CIKM)*. ACM, 1001–1010.

[22] Xiang Liu, Yuchun Guo, Yishuai Chen, and Xiaoying Tan. 2018. Trajectory Privacy Protection on Spatial Streaming Data with Differential Privacy. In *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–7.

[23] Yiwen Nie, Liusheng Huang, Zongfeng Li, Shaowei Wang, Zhenhua Zhao, Wei Yang, and Xiaorong Lu. 2016. Geospatial streams publish with differential privacy. In *International Conference on Collaborative Computing: Networking, Applications and Worksharing*. Springer, 152–164.

[24] Mateusz Pawlik, Thomas Hütter, Daniel Kocher, Willi Mann, and Nikolaus Augsten. 2019. A link is not enough–reproducibility of data. *Datenbank-Spektrum* 19, 2 (2019), 107–115.

[25] Xuebin Ren, Liang Shi, Weiren Yu, Shusen Yang, Cong Zhao, and Zongben Xu. 2022. LDP-IDS: Local Differential Privacy for Infinite Data Streams. *Proceedings of the 2022 SIGMOD International Conference on Management of Data* (2022), 1064—-1077.

[26] Qian Wang, Xiao Lu, Yan Zhang, Zhibo Wang, Zhan Qin, and Kui Ren. 2016. Secweb: Privacy-preserving web browsing monitoring with w-event differential privacy. In *Proceedings of the 12th EAI International Conference on Security and Privacy in Communication Systems (SecureComm)*. Springer, 454–474.

[27] Qian Wang, Yan Zhang, Xiao Lu, Zhibo Wang, Zhan Qin, and Kui Ren. 2016. Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy. *IEEE Transactions on Dependable and Secure Computing (TDSC)* 15, 4 (2016), 591–606.

[28] Qian Wang, Yan Zhang, Xiao Lu, Zhibo Wang, Zhan Qin, and Kui Ren. 2016. RescueDP: Real-time spatio-temporal crowd-sourced data publishing with differential privacy. In *Proceedings of the 35th Annual IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 1–9.

[29] Tianhao Wang, Joann Qiongna Chen, Zhikun Zhang, Dong Su, Yueqiang Cheng, Zhou Li, Ninghui Li, and Somesh Jha. 2021. Continuous release of data streams under both centralized and local differential privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 1237–1253.

[30] Teng Wang, Xinyu Yang, Xuebin Ren, Jun Zhao, and Kwok-Yan Lam. 2019. Adaptive differentially private data stream publishing in spatio-temporal monitoring of IoT. In *Proceedings of the 38th IEEE International Performance Computing and Communications Conference (IPCCC)*. IEEE, 1–8.

[31] Zhibo Wang, Xiaoyi Pang, Yahong Chen, Huajie Shao, Qian Wang, Libing Wu, Honglong Chen, and Hairong Qi. 2018. Privacy-preserving crowd-sourced statistical data publishing with an untrusted server. *IEEE Transactions on Mobile Computing* 18, 6 (2018), 1356–1367.

[32] Nico Weidmann. 2019. Differentially Private Event Sequences over Infinite Streams.

[33] Greg Welch and Gary Bishop. 1995. *An introduction to the Kalman filter*. Technical Report. Department of Computer Science, University of North Carolina at Chapel Hill.

[34] Jiajun Zhang, Xiaohui Liang, Zhikun Zhang, Shibo He, and Zhiguo Shi. 2017. Re-DPoctor: Real-time health data releasing with w-day differential privacy. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–6.

KIT Scientific Working Papers
ISSN 2194-1629

www.kit.edu