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 Salzburg
Salzburg, Austria
martin.schaeler@sbg.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 are 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 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.

PVLDB Reference Format:
Christine Schäler, Thomas Hütter, and Martin Schäler. Benchmarking the Utility of w-event Differential Privacy Mechanisms – When Baselines Become Mighty Competitors. PVLDB, 16(8): 1830 - 1842, 2023.
doi:10.14778/3594512.3594515

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://dbresearch.uni-salzburg.at/projects/dpbench.

1 INTRODUCTION

Monitoring data streams continuously facilitates new analysis tasks, e.g., controlling real-time intelligent traffic [20] or electricity distribution systems [3]. However, the privacy requirements of the data owners have to be fulfilled to deploy the tasks. To ensure strong privacy for streams, the w-event differential privacy (DP) framework [25] is the state-of-the-art. It gives a provable 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 DP [9, 10, 27, 29, 36, 38]. 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., sparsity [38]. Unfortunately, there are little insights on which mechanisms provide high data utility for which stream properties, mainly due to incomparable empirical studies. This imposes a 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 studies 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 streams and competitors. This hampers the comparison of existing results. Unfortunately, guidelines for empirical studies on static data [21] (e.g., finite time series [19] or relational databases [7]) 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 research on novel mechanisms.

Limitations of Existing Studies. The comparability of empirical studies on w-event DP is limited by inconsistent experimental elements, e.g., the selection of data streams, mechanisms, and error metrics indicating a mechanism's utility.

Specifically, most studies focus on a small set of real-world streams [9, 10, 27, 29, 36, 38]. We observe two limitations: First, many streams are not publicly available. Second, relevant preprocessing steps are unknown or differ highly [8, 31]. Studies using artificial data to investigate the influence of data properties exist only for static data [21] and finite time series [19]. Analyzing available streams indicates that they are often sparse, i.e., mainly contain zero values. Thus, releasing the same value all the time yields good utility, as performed by one of the baseline mechanisms [25].

Quantifying the benefits that data administrators can achieve from the latest w-event mechanism is impossible since many studies do not compare to both baseline mechanisms. Hence, it is hard to decide whether a baseline suffices the use case or a sophisticated mechanism is needed. Moreover, state-of-the-art mechanisms are

Proceedings of the VLDB Endowment, Vol. 16, No. 8 ISSN 2150-8097.
doi:10.14778/3594512.3594515

Table 1: Location monitoring stream S. Database D_t contains the location of all individuals at timestamp t. Query $Q(D_t)$ contains the number of individuals per location.

Ind.	D_1	D_2	D_3	...
Axel	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	...

highly complex and subtle differences in the implementation or parameters can have a significant effect on a mechanism's utility. This is a serious limitation, since implementations of mechanisms are rarely publicly available. Consequently, verifying the experimental results of re-implemented mechanisms is virtually impossible.

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

Benchmark requirements. Based on a comprehensive literature survey, we identify that all existing empirical studies on w-event mechanisms can be described by four elements: mechanisms, streams, privacy requirements, and utility metrics. For each element, we outline the limitations of prior studies and propose requirements to ensure comparable results. Moreover, our survey reveals that 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.

Empirical study and new insights. We conduct the largest empirical evaluation of w-event DP mechanisms so far. It is based on our benchmark, comprising of 259,000 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 background on w-event differential privacy, the w-event mechanism framework, and common utility metrics.

2.1 w-Event Differential Privacy

The w-event differential privacy (DP) framework is the current standard for ensuring differential privacy of continuously computed aggregate queries on streams. Rather than protecting the stream entirely, which requires an infinite amount of noise, one protects every running window of at most w timestamps [25].

Let $S = (D_1, D_2, ..)$ be a stream collecting database D_t at timestamp t as shown in Table 1. Each row in D_t corresponds to an individual data owner and each column to an activity, e.g., location visit. A query of interest Q is the number of data owners per location at each timestamp. This query is a multi-dimensional count query (i.e., histogram) with one count per location and timestamp. All DP 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., data owner [12]. Further, let $S_p = (D_1, \dots, D_p)$ be a stream prefix of length p. Intuitively, two stream prefixes are w-*neighbors* if (1) the databases collected at each timestamp are the same or neighbors, and (2) all neighboring databases fit into a window of size w (cf. Definition 1).

Definition 1 (w-Neighboring Stream Prefixes [25]). *Let w be a positive integer, p the length of the stream prefixes, 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 $D_t \neq D'_t$ and
(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 $D_{t_2} \neq D'_{t_2}$ holds.

The desired *privacy level* ϵ usually lies between 0.1 and 1. A smaller value means better privacy. From Definition 2, w-event DP 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}. Consequently, in the w-event DP framework, the data owners specify their privacy requirements in terms of a (ϵ,w)-tuple via a decision by consensus.

Definition 2 (w-Event ϵ-Differential Privacy [25]). *Let $\mathcal{M}$ be a non-deterministic mechanism that takes a stream prefix S_p of arbitrary size as input and outputs a transcript R of S_p, i.e., a sanitized query result. Further, let Range($\mathcal{M}$) be the set of all possible outputs of $\mathcal{M}$. 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 $Pr[\mathcal{M}(S_p) = R] \leq e^{\epsilon} \cdot Pr[\mathcal{M}(S'_p) = R]$.*

A DP mechanism for numeric queries usually adds noise based on the zero-mean Laplace distribution Lap(λ) to each of the *dim* outputs of a query $Q : D_t \rightarrow \mathbb{R}^{\text{dim}}$, e.g., histogram bins. The scale $\lambda = \frac{\Delta Q}{\epsilon}$ depends on the privacy budget ϵ and the *global sensitivity* $\Delta Q = \max_{D_t, D'_t} ||Q(D_t) - Q(D'_t)||_1$ at any possible timestamp t. 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. The only premise is that the budget spend by these mechanisms does not exceed ϵ for every rolling window of size w (cf. Theorem 1).

Theorem 1 (Composition [25]). *Let $\mathcal{M}$ be a mechanism processing a stream prefix $S_p = (D_1, \dots, D_p)$ and outputting a transcript of released values $R = (r_1, \dots, r_p)$. Assume that we can decompose $\mathcal{M}$ into p sub-mechanisms $\mathcal{M}_1, \dots, \mathcal{M}_p$, 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 ϵ-differential privacy if*

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

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

Input (ϵ, w): privacy requirements, D_t: database at timestamp t, Q: query, l: last sampling time
Output r_t: sanitized query result at time t

```
function $\mathcal{M}_t((\epsilon, w), D_t, Q, l)$
    if isSamplingPoint$(\epsilon, w, D_t, l)$ then
        $\epsilon_t \leftarrow$ budgetAllocation$(\epsilon, w, D_t, l)$
        $p_t \leftarrow$ pertubation$(\epsilon_t, Q, D_t)$
        $r_t \leftarrow$ filtering$(p_t)$                ▷ sanitized query result
        $l \leftarrow t$
    else $r_t \leftarrow r_l$                            ▷ approximation
    return $r_t$
```

A sub-mechanism $\mathcal{M}_t$ has five inputs[1]: privacy requirements (ϵ, w), database D_t, query Q, and the last timestamp l where a sub-mechanism released a *sanitized* query result. Note that not at all timestamps the query result is sanitized, but previously sanitized query results can be released again. The output of the sub-mechanism is the released query result r_t. Intrinsically, the sub-mechanism implements four functions that are described below. 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.

Perturbation-Function. Mechanisms calculate the true query result $Q(D_t)$ before perturbing using the allocated budget. Note that all identified mechanisms perturb using Laplace noise.

Filtering-Function. The post-processing immunity of differential privacy [15] 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.

Table 2: Computation of the functions in the w-event mechanism framework for the baselines Uniform and Sample [25].

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

2.3 Utility Metrics

To measure the utility of the released stream, researchers have two options: First, quantifying 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)* [6, 18, 25, 27, 37, 40]:

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

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

$\gamma > 0$ is a sanity bound to mitigate the effect of small query results. The second option is to quantify the accuracy of the streams in context of an analysis task, like forecasting, event monitoring, or anomaly detection. To this end, one relies on a task-specific metric, like the AUC-ROC score [34]. It quantifies the area under the ROC-curve illustrating the relation between the true-positive (TPR) and false-positive (FPR) rate. A score of 1 indicates prefect anomaly detection, 0 inverses the labels, and 0.5 has no detection quality.

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 results. We identify these elements by conducting a comprehensive literature survey with the following methodology: We include all publications that perform an experimental evaluation on *streams*, i.e., not only finite time series which excludes [2, 11], and are published at notable peer-reviewed venues, e.g., VLDB, SIGMOD, and CCS. In total, we include 16 publications listed in Table 3. As a result of our survey, we formalize the requirements of an empirical study on w-event mechanisms as a 4-tuple $(\mathbb{M}, \mathbb{S}, \mathbb{P}, \mathbb{E})$:

- $\mathbb{M}$ 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 detail and introduce requirements that ensure the comparability of empirical studies. Based on the requirements, we reveal the limitations of existing studies.

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 extent previous works address these requirements (summarized in Table 3).

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

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*, d *dimension*, n the block size in (w,n)-DP [28]. Note that ($\mathbb{M}$-R4) and ($\mathbb{M}$-R5) are not applicable.

Reference	Model	Granu- larity	($\mathbb{M}$-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$_w$ [25]	cent.	w-event	✓	both	✓[a]	-	([40,200], 1)	MAE, MRE $\gamma = u$
Retroact. Group. [7]	cent.	event	✓	Uniform	✗	([0.02,0.1], 1)	n.a.	MAE, MRE $\gamma = u$
DSAT$_w$ [27]	cent.	w-event	✓	none	✗	([0.5,1], 800)	([200,1000], u)	total sum of squared error
SecWeb [36]	cent.	w-event	✓	Uniform	✗	([0.01,1], 120)	([40,240], 1)	MAE, MRE $\gamma = 1$
G-event [10]	cent.	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 [29]	cent.	w-event	✓	Uniform	✗	({0.5,1.0}, 1)	({10,50}, u)	MRE $\gamma = u$
RescueDP [37, 38]	cent.	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 [44]	cent.	w-day	✗	none	✗	([0.5,1.5], 14)	([7,35], 1)	MAE, MRE $\gamma = 0.05\% \cdot \sum_{t=1}^{p} Q(D_t)$
PeGaSuS [9]	cent.	event	✓	Uniform	✗	({0.01, 0.1}, 1)	n.a.	MAE, TPR, ROC of event monitoring
Local DP [16]	local	w-event	✓	none	✗	({1.1,1.9}, 4)	([10,100], 1)	MAE, RMSE
STBD [28]	cent.	(w, n)	✓	Uniform	✗	([0.2,1.0], 120)	([40,200], 1)	MAE
DPS [17]	local	w-event	✗	Uniform	✗	-	([0.01, 1], u)	unspecified 'average error'
AdaPub [40]	cent.	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 [41]	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 [39]	cent.	event	✓	none	✗	([0.01,0.5], 1)	n.a.	mean squared error
LPD-IDS [33]	local	w-event	✓	both	✗	({0.5, 2.5}, 20)	({10, 50}, 1)	MRE $\gamma_d = u$, ROC of event monitoring

[a] FAST used for FAST$_w$: http://www.mathcs.emory.edu/~lxiong/aims/FAST/

($\mathbb{M}$-R1) Proofing the Desired Privacy Definition. A fair comparison of mechanisms requires identical privacy definitions. Considering DP, the privacy definition consists of the *model* and the *granularity*. There exist three models: In the *centralized model* [12], a *trusted* data administrator collects all rows of the stream S to calculate and sanitize $Q(S)$. In the *local model* [23], each individual stream (i.e., row in S) is sanitized such that it fulfills local DP. Consequently, an *untrusted* data administrator calculates $Q(S)$. In the *distributed model* [1], each individual stream is sanitized with Gamma noise and encrypted. The untrusted data administrator decrypts the summed-up stream that satisfies DP. Mechanisms for different models are not comparable, because complying with the local or distributed model provides a lower utility than the centralized model. The most commonly used mechanism granularities (cf. Table 3) are event-level [14] and w-event [25]. Note that w-event generalizes event-level. Mechanisms with different granularities are comparable if the mechanism can be parameterized to provide the desired granularity. For instance, an event-level DP mechanism satisfies w-event DP if we provide a budget of $\epsilon_t = \frac{\epsilon}{w}$ per timestamp t [25].

In order to include a mechanism in a benchmark, the authors of the mechanism need to (1) proof that the claimed definition is satisfied and, if applicable, (2) state how the mechanism can be parameterized such that it achieves the desired granularity. 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) Including both Baseline Mechanisms. Two baseline mechanisms, i.e., Uniform and Sample, are proposed in the original w-event DP publication [25]. 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. The perturbation error 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 is defined by the difference between the true query result $Q(D_t)$ and the last released sanitized result r_l. If the query result fluctuation is small, the approximation error is also small.

Mechanism Uniform samples every timestamp by allocating $\epsilon_t = \frac{\epsilon}{w}$ budget for perturbation; hence, only a perturbation error is introduced. By contrast, mechanism 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, e.g., the sampling decisions of multiple mechanisms rely on a *proportional-integral-derivative (PID) controller* [4, 19, 27, 38] or Kalman filter [19, 38, 42]. We observe that minor differences in the implementation or parameters can have a significant effect on a mechanism's utility, e.g., rounding the query result in the Filtering function. Hence, we advocate to make implementations publicly available to provide additional insights and facilitate the comparison. Our survey reveals that only one out of 16 publications provide access to their implementation.

($\mathbb{M}$-R4) Private Parameter Determination. Parameters of a mechanism that are used on the true stream need to be computed in a private way [21], e.g., the number of rows which is private information. None of our surveyed publications addresses 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 mechanisms do not fulfill this requirement. To solve this issue, we suggest to follow the proposal of [21] to use *mechanism repair functions.*

($\mathbb{M}$-R5) Homogeneity of Background Knowledge. Most mechanisms use components, like PID controllers [4], 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 [21].

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 [33]. Even though related work [18, 25, 40] 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 & Reproducible Preprocessing. Our literature survey reveals that most approaches focus on real-world streams from specific use cases. Even though multiple publications use the same streams, the respective study results are not necessarily comparable. The reason is that the streams are preprocessed differently as can be seen in the following example on the WorldCup dataset: Its raw data contains the logs of 89,997 websites. [25] refers to all 89,997 websites while [38] samples 2,000 of them, leading to inconsistent utilities values. 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 [8, 31].

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. In all publications listed in Table 3, two types of experiments are conducted:

($\mathbb{P}$ - R1) Vary-ϵ. Effects of ϵ for a fixed value of w. Selecting an appropriate ϵ value is an ongoing research line [13, 24, 26, 30]. In most studies, ϵ is varied between 0.1 and 10.

($\mathbb{P}$ - R2) Vary-w. Effects of w for a fixed value of ϵ, mostly $\epsilon = 1$. Note that 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.

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

Elem.	Instantiation
$\mathbb{M}$	(1) Baselines: Sample [25], Uniform [25]; (2) Competitors: FAST$_w$ [25], DSAT$_w$ [27], BD [25], BA [25], RescueDP [37, 38], AdaPub [40], PeGaSuS [9]
$\mathbb{S}$	(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 $\gamma_d = 0.1\% \cdot \sum_{t=1}^{p} Q(D_t)[d]$ over 100 runs (3) Analysis task anomaly detection: average AUC-ROC score

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, [38] uses a data-dependent sanity bound γ, whereas [10] fixes $\gamma = 1.0$. In three publications, the sanity bound is not stated, even though the used streams contain query results of 0, requiring $\gamma > 0$. Only two works [9, 33] quantify the utility for a specific analysis task (here: event monitoring) by using a task-specific metric.

4 BENCHMARK DEFINITION

We now introduce a benchmark for w-event DP mechanisms, aiming at the comparability of experimental results. Based on Table 3, we focus on the frequently used centralized model. 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 259,000 single experiments, i.e., mechanism runs. We discuss how to meet the requirements of each element and argue how to ensure the validity and comprehensiveness of the results.

4.1 Mechanism Set $\mathbb{M}$

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

($\mathbb{M}$-R1)-($\mathbb{M}$-R2) Considered Mechanisms. We include the baseline mechanisms Sample and Uniform, as well as *all* mechanisms found in our literature study that are either proven to support w-event DP directly or can be parameterized such that they achieve w-event DP. According to Table 3, this applies to FAST$_w$ [25], DSAT$_w$ [27], RGP [29], SecWeb [36], RescueDP [37, 38] and AdaPub [40]. Since SecWeb is a prequel of RescueDP, we do not include SecWeb in our benchmark. We also exclude RGP, since it is only applicable to hierarchic location count streams. We further include all mechanisms used as competitors for an included mechanism. This includes PeGaSuS [9] 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.

($\mathbb{M}$-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. DSAT$_w$ uses the number of rows in the stream. Since streams that feature a different number of rows might be 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 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 [27].

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 [25] 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 [25]. 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 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, i.e., a *desired sampling rate* of 15% in FAST$_w$ and DSAT$_w$, and the parameters of the PID controller. While the publications proposing RescueDP [37, 38] and FAST [19] suggest the same PID parameters, the values used in DSAT$_w$ [27] differ due to a different operational purpose of the PID controller. As a result, we use the parameters as 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 follow four key principles: *(a) Favor original implementations:* Unfortunately, this only holds for one mechanism, namely FAST$_w$ (cf. Table 3). *(b) Re-use of well-known mechanism parts:* Multiple mechanisms use the same components (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. *(c) Consistency checks:* All mechanisms are implemented redundantly and independently by up to three people, eventually leading to consistent results. *(d) Contact original authors:* Some of our results highly deviate from the results in the original publication. Consequently, we contacted the original authors of w-event DP [25] and thankfully received re-implementations of the baselines and major parts of BD and BA. This ensures the correctness of all baselines and major parts of BD and BA, and allowed us to reuse identical mechanism parts.

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

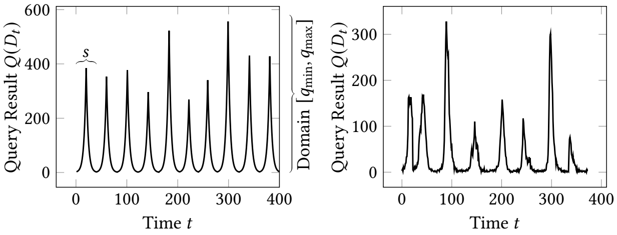
Stream	Avail.	Referenced in	Limitations
APASCologne	✓	[17]	-
DNS	✗	[39]	-
Fare	✓	[39]	raw data only
Flu Death	✓	[40]	different season
Flu Outpatient	✓	[18]	different ages & years
Foursquare	✓	[33]	-
GeoLife	✗	[28]	-
Heart rates	✗	[44]	-
Kosarak	✓	[39]	raw data only
Montreal traffic	✗	[7]	-
Nice ride	✓	[41]	-
POS	✗	[39]	-
Retail	✓	[40]	-
Rome traffic 1	✗	[25]	-
Rome traffic 2	✗	[29]	-
San Joaquin	✓	[27, 37, 38]	data generator only
State Flu	✓	[40]	-
TDrive	✓	[27, 33]	-
Taobao	✓	[33]	requires account
Taxi Porto	✓	[27, 28, 37, 38, 41]	raw data only
Traffic Seattle	✗	[18]	-
Unemployment	✓	[18]	-
US census	✓	[27]	raw data only
WiFi traces 1	✗	[9]	-
WiFi traces 2	✗	[16]	-
WorldCup	✓	[7, 10, 25, 36–38]	raw data only

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 streams (i.e., $\mathbb{S}$-R1). Second, we conduct experiments on a comprehensive set of data streams used in literature (i.e., $\mathbb{S}$-R2).

($\mathbb{S}$-R1) Artificial Streams Reflecting Stream Properties. The intention behind using artificial streams is to study the influence of relevant stream properties on a mechanism's utility in a structured way. Generating meaningful artificial streams 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 [22]. 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 stream generator that allows to investigate each of the properties in isolation.

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



(a) Artificial stream. **(b) Flu Death stream.**

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

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 any negative perturbed query results to zero. Hence, whenever p_t is negative after adding the Laplace noise (e.g., -10), the mechanism releases the true query result instead. This occurs e.g., for any $Q(D_t) = q_{\min} = 0$ in half of the cases. 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 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 $a = q_{\max} - q_{\min}$. 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 streams listed in Table 5. Therefore, we do not consider this property.

Data Generator. Figure 1a shows a stream produced by our generation algorithm (cf. Algorithm 2). Generally, the artificial streams 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 Gaussian distribution $\mathcal{G}(s = 40, 2)$. 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. We also mimic inter-seasonal downtime by dicing the season minimum with $\mathcal{G}(s = 8, 2)$, i.e, some query result close to zero. Since the maximum query result of the stream generated this way depends on the actual length of the season and the diced minimal query result, we need to normalize the maximum query result 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

```
Input p: stream length, s: average season length, a: maximal amplitude
Output Q(D_1), .., Q(D_p): Query result stream of p timestamps
function generateStream(p, s, a)
    $t \leftarrow 1, e \leftarrow 1.5$
    while $t < p$ do                                   ▷ Each loop generates one season
        $\text{sl} \leftarrow \mathcal{G}(s, 2)$                           ▷ Dice season length
        $\text{val} \leftarrow \mathcal{G}(8, 2)$                ▷ Dice season minimum, close to 0
        $Q(D_t) \leftarrow \text{val}; t{+}{+}$
        for $i = 1$ to $\text{sl}/2$ do
            $val \leftarrow e \cdot Q(D_{t-1})$                    ▷ Exponential growth
            $Q(D_t) \leftarrow \text{val}; t{+}{+}$
        ...                                          ▷ Symmetric shrinking phase
    $\max \leftarrow \max\{Q(D_1), .., Q(D_{p-1})\}$
    for $i = 1$ to $p$ do
        $Q(D_i) \leftarrow Q(D_i)/\max \cdot a$             ▷ Ensure desired amplitude
    return $Q(D_1), ..., Q(D_p)$                         ▷ Ensure correct length
```

($\mathbb{S}$-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 (cf. Table 5). 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 publicly available[2].

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 $\epsilon \in [0.1, 1]$. For the vary-w experiments, we use $\epsilon = 1$ like most studies. We vary $w \in [40, 200]$ with a w increment of 40, s.t. there is an overlap with various other studies. When measuring utility of anomaly detection, we show the results for $w \in [1, 32]$. The rational is that the results remain constant for larger values.

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

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

each experiment 100 times and use the average of the utility metrics. We use both types of metrics introduced in Section 2.3. First, we use MAE and MRE with $\gamma_d = 0.1\% \cdot \sum_{t=1}^{p} Q(D_t)[d]$ for dimension d. Besides the average error, we quantify the variance of the error as suggested by [21] 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. Second, we select anomaly detection as analysis task where we use the AUC-ROC score [34] as task-specific metric.

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

We aim at understanding the effects of the identified stream properties seasonal period length s and amplitude a on a mechanism's utility using artificial streams. 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, amplitude a, and privacy requirements (ϵ, w)?

For brevity, we subsequently focus on the average MAE, short MAE, to answer these questions since the result patterns for the 0.95 quantile of MAE and MRE are similar[3]. 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 $c = (m \in \mathbb{M}, (s, a), (\epsilon, w) \in \mathbb{P})$. The MAE deterioration compares the MAE of mechanism m to the 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 the influence of the stream properties (a, s) and privacy requirements (ϵ, w) on 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 releasing 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 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 $\epsilon = 1$. Summarizing, mechanisms provide a high utility either for small or for large amplitudes independent of other parameters such as s, ϵ, or w.

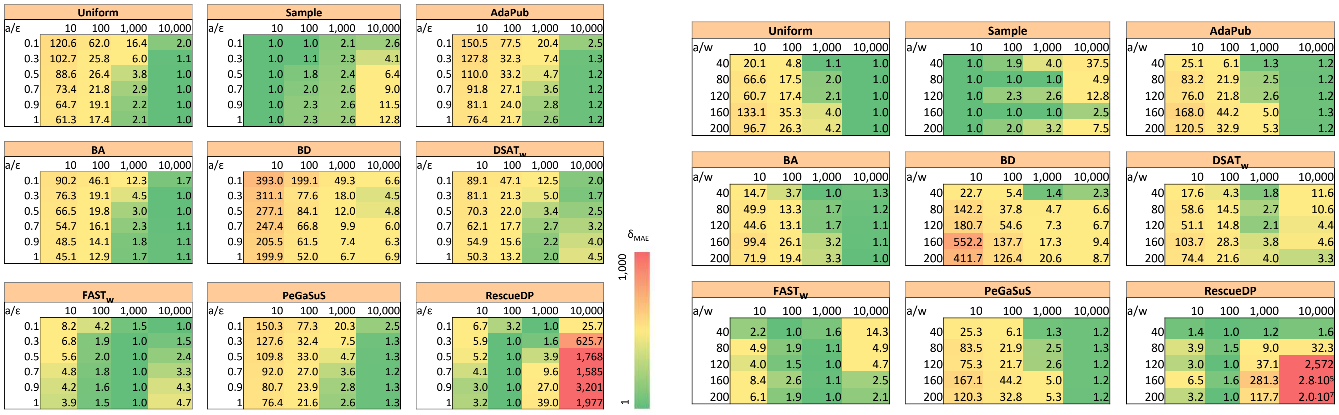
5.1.2 *(RQ2) Can we recommend a mechanism for specific stream properties (s, a) and privacy requirements (ϵ, w)?*

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.

Uniform Supremacy. Our results suggest that mechanism Uniform is among the best for large amplitudes $a \geq 1,000$ and non-restrictive 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, AdaPub and PeGaSuS are expected to 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 is 11 with $a = 10$, $w = 100$, and $\epsilon = 1$, whereas Uniform's bound is 100, i.e., 10 times larger. However, we rarely observe Sample's bound and the observed MAE is several factors smaller. The rational is that Sample has a tendency to release the most frequent query results very accurately.

[3] This also holds for other metrics, e.g., the L2 error.



Uniform

a/ε	10	100	1,000	10,000
0.1	120.6	62.0	16.4	2.0
0.3	102.7	25.8	6.0	1.1
0.5	88.6	26.4	3.8	1.0
0.7	73.4	21.8	2.9	1.0
0.9	64.7	19.1	2.2	1.0
1	61.3	17.4	2.1	1.0

Sample

a/ε	10	100	1,000	10,000
0.1	1.0	1.0	2.1	2.6
0.3	1.0	1.1	2.3	4.1
0.5	1.0	1.8	2.4	6.4
0.7	1.0	2.0	2.6	9.0
0.9	1.0	2.3	2.6	11.5
1	1.0	2.3	2.6	12.8

AdaPub

a/ε	10	100	1,000	10,000
0.1	150.5	77.5	20.4	2.5
0.3	127.8	32.3	7.4	1.3
0.5	110.0	33.2	4.7	1.2
0.7	91.8	27.1	3.6	1.2
0.9	81.1	24.0	2.8	1.2
1	76.4	21.7	2.6	1.2

BA

a/ε	10	100	1,000	10,000
0.1	90.2	46.1	12.3	1.7
0.3	76.3	19.1	4.5	1.0
0.5	66.5	19.8	3.0	1.0
0.7	54.7	16.1	2.3	1.1
0.9	48.5	14.1	1.8	1.1
1	45.1	12.9	1.7	1.1

BD

a/ε	10	100	1,000	10,000
0.1	393.0	199.1	49.3	6.6
0.3	311.1	77.6	18.0	4.5
0.5	277.1	84.1	12.0	4.8
0.7	247.4	66.8	9.9	6.0
0.9	205.5	61.5	7.4	6.3
1	199.9	52.0	6.7	6.9

DSAT$_w$

a/ε	10	100	1,000	10,000
0.1	89.1	47.1	12.5	2.0
0.3	81.1	21.3	5.0	1.7
0.5	70.3	22.0	3.4	2.5
0.7	62.1	17.7	2.7	3.2
0.9	54.9	15.6	2.2	4.0
1	50.3	13.2	2.0	4.5

FAST$_w$

a/ε	10	100	1,000	10,000
0.1	8.2	4.2	1.5	1.0
0.3	6.8	1.9	1.0	1.5
0.5	5.6	2.0	1.0	2.4
0.7	4.8	1.8	1.0	3.3
0.9	4.2	1.6	1.0	4.3
1	3.9	1.5	1.0	4.7

PeGaSuS

a/ε	10	100	1,000	10,000
0.1	150.3	77.3	20.3	2.5
0.3	127.6	32.4	7.5	1.3
0.5	109.8	33.0	4.7	1.3
0.7	92.0	27.0	3.6	1.2
0.9	80.7	23.9	2.8	1.3
1	76.4	21.6	2.6	1.3

RescueDP

a/ε	10	100	1,000	10,000
0.1	6.7	3.2	1.0	25.7
0.3	5.9	1.0	1.6	625.7
0.5	5.2	1.0	3.9	1,768
0.7	4.1	1.0	9.6	1,585
0.9	3.0	1.0	27.0	3,201
1	3.2	1.0	39.0	1,977

(a) Vary-ϵ experiments with $w = 120$.

Uniform

a/w	10	100	1,000	10,000
40	20.1	4.8	1.1	1.0
80	66.6	17.5	2.0	1.0
120	60.7	17.4	2.1	1.0
160	133.1	35.3	4.0	1.0
200	96.7	26.3	4.2	1.0

Sample

a/w	10	100	1,000	10,000
40	1.0	1.9	4.0	37.5
80	1.0	1.0	1.0	4.9
120	1.0	2.3	2.6	12.8
160	1.0	1.0	1.0	2.5
200	1.0	2.0	3.2	7.5

AdaPub

a/w	10	100	1,000	10,000
40	25.1	6.1	1.3	1.2
80	83.2	21.9	2.5	1.2
120	76.0	21.8	2.6	1.2
160	168.0	44.2	5.0	1.3
200	120.5	32.9	5.3	1.2

BA

a/w	10	100	1,000	10,000
40	14.7	3.7	1.0	1.3
80	49.9	13.3	1.7	1.2
120	44.6	13.1	1.7	1.1
160	99.4	26.1	3.2	1.1
200	71.9	19.4	3.3	1.0

BD

a/w	10	100	1,000	10,000
40	22.7	5.4	1.4	2.3
80	142.2	37.8	4.7	6.6
120	180.7	54.6	7.3	6.7
160	552.2	137.7	17.3	9.4
200	411.7	126.4	20.6	8.7

DSAT$_w$

a/w	10	100	1,000	10,000
40	17.6	4.3	1.8	11.6
80	58.6	14.5	2.7	10.6
120	51.1	14.8	2.1	4.4
160	103.7	28.3	3.8	4.6
200	74.4	21.6	4.0	3.3

FAST$_w$

a/w	10	100	1,000	10,000
40	2.2	1.0	1.6	14.3
80	4.9	1.9	1.1	4.9
120	4.0	1.5	1.0	4.7
160	8.4	2.6	1.1	2.5
200	6.1	1.9	1.0	2.1

PeGaSuS

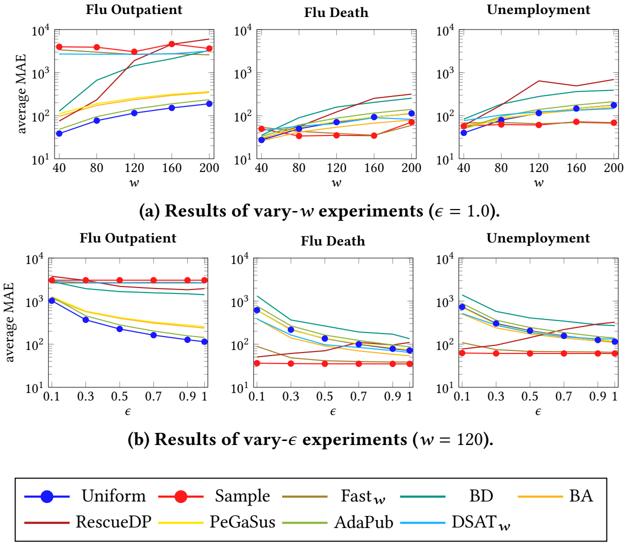
a/w	10	100	1,000	10,000
40	25.3	6.1	1.3	1.2
80	83.5	21.9	2.5	1.3
120	75.3	21.7	2.6	1.2
160	167.1	44.2	5.0	1.2
200	120.3	32.8	5.3	1.2

RescueDP

a/w	10	100	1,000	10,000
40	1.4	1.0	1.2	1.6
80	3.9	1.5	9.0	32.3
120	3.0	1.0	37.1	2,572
160	6.5	1.6	281.3	$2.8{\cdot}10^5$
200	3.2	1.0	117.7	$2.0{\cdot}10^7$

(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 $s = 80$.



(a) Results of vary-w experiments ($\epsilon = 1.0$).

(b) Results of vary-ϵ experiments ($w = 120$).

Figure 3: Average error of vary-w and vary-ϵ experiments for one-dimensional real-world streams. ○ marks the baselines.

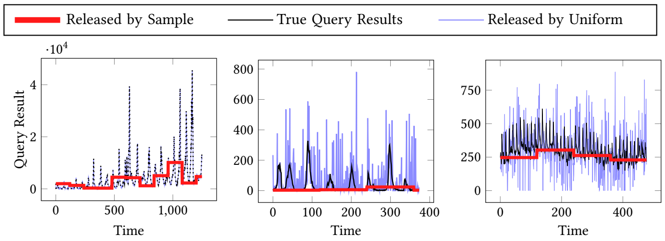
Hypersensitive Data-Adaptive Sampling. The small MAEs of Sample for small amplitudes and restrictive privacy requirements suggest that the perturbation error needs to be minimized via sampling. The mechanisms BD, BA, DSAT$_w$, FAST$_w$, and RescueDP feature data-adaptive sampling. The idea is to invest a fraction of the budget ϵ to monitor the stream. In case the mechanism monitors a large enough change, a new query result is released. 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 since data-adaptive sampling features a hypersensitivity for small changes in the query result. We observe the following tendencies: In case the growing phase of a new season starts, the initial small changes of the query result are well reflected. In addition, there is a sampling timestamp close to the peak of the first season. Thereby, a large fraction of the 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 results are produced in the shrinking phase incurring a high MAE. That becomes worse when 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

We evaluate the results on one-dimensional real-world streams with two objectives: First, verifying whether the results of real-world and artificial streams are consistent, particularly the baseline supremacy. Second, understanding the abstract error measures (e.g., MAE) in context of the data streams. In a nutshell, our key findings are that the results on real-world and artificial streams are consistent and common error metrics are not well-suited for streams.

5.2.1 Confirmation of Micro Benchmark Results. Figure 3 depicts MAE for all mechanisms and one-dimensional real-world streams.

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 negative noise, a count of 0 is released. 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



(a) Flu Outpatient
MAE Uniform: 115.3
MAE Sample: 3, 056

(b) Flu Death
MAE Uniform: 76.3
MAE Sample: 34.7

(c) Unemployment
MAE Uniform: 115.6
MAE Sample: 60.8

Figure 4: True query result and exemplary releases of the baselines Uniform and Sample for $\epsilon = 1.0$ and $w = 120$.

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 Abstract Utility Metric Values. 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, e.g., Sample's utility appears to be almost independent of the privacy requirements. Thus, we examine the semantics of the abstract error values by considering common applications performed on 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. Thus, we examine the sanitized query results of Sample and Uniform w.r.t. seasonality and level. At least one mechanism is ensured to provide a good MAE for every stream due to the baseline supremacy. Our explanations hold in general and are based on exemplary sanitized releases, like the real-world streams shown in Figure 4.

Maintaining Seasonal Growing & Shrinking of the Stream. Sample erases seasonality independent of the observed MAE, even for streams where Sample performs best (cf. Figure 4). In case the mechanism only samples once 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.

Uniform maintains the seasonality well when amplitudes are large compared to the introduced noise. As the expected noise is $\frac{w}{\epsilon}$, the amplitude decides 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 observed in Figure 4a but not in 4b. Hence, MAE has no meaning for maintaining seasonality. However, there is a relation between MAE and the level maintained by Uniform, as we discuss next.

Maintaining Level & Amplitude of the Stream. Recap that the true level of the stream is defined by q_{min} and the true amplitude $a = q_{\text{max}} - q_{\text{min}}$. The level and amplitude of the sanitized stream released by Uniform depend on the true level and amplitude. In case $q_{\text{min}} > \frac{w}{\epsilon}$, i.e., the level is higher than the expected noise, the sanitized stream released by Uniform features the domain $[q_{\text{min}} - \frac{w}{\epsilon}, q_{\text{max}} + \frac{w}{\epsilon}]$. This can be observed in Figure 4c 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 4b, i.e., the minimum possible value of a count query. Thus, truncating count queries lead to an expected level change of $[\max(q_{\text{min}} - \frac{w}{\epsilon}, 0), q_{\text{max}} + \frac{w}{\epsilon}]$.

Since Sample has a low perturbation error, the domain is not enlarged, i.e., only values within the original minimum and maximum values are released. However, the sanitized streams usually miss the seasonal peaks of the true streams. Large MAE values, specifically above 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.

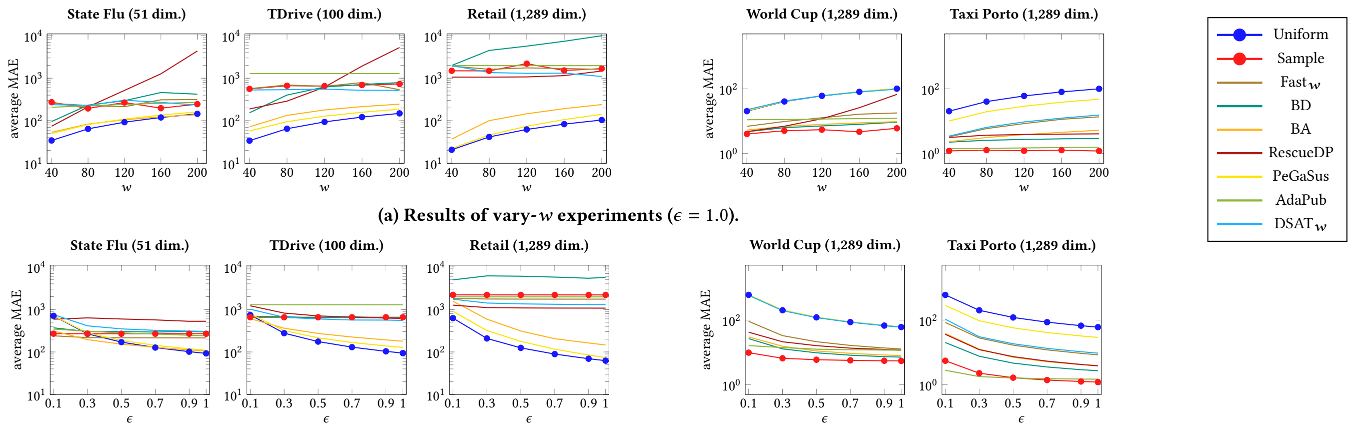
5.3 Multi-dimensional Real-World Streams

We now present the results (cf. Figure 5) of the multi-dimensional streams from 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.

Adaptive dimension-grouping finds 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. The mechanisms perturb the sum of the query results over all dimensions in group g and then assign each dimension the average of the perturbed sum. This reduces the expected perturbation error from $\frac{1}{\epsilon_t}$ to $\frac{1}{\epsilon_t \cdot |g|}$ [37]. Hence, with increasing dimensionality the perturbation error is highly reduced.

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. The table contains the query result distribution of the true query results over all dimensions of the preprocessed streams. We observe that q_{min} of TaxiPorto and WoldCup equals zero and the 90% quantile of $Q(D_t)$ is very small compared to the other streams.

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.



(a) Results of vary-w experiments (ϵ = 1.0).

(b) Results of vary-ϵ experiments (w = 120).

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

Table 6: Properties of multi-dimensional streams.

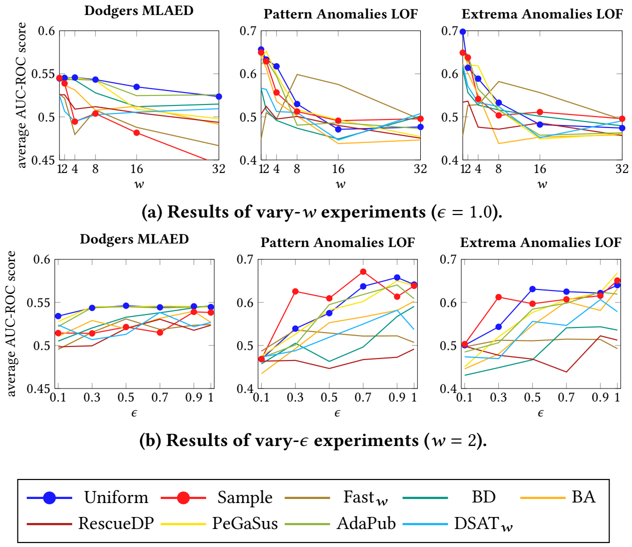
Stream S	dim	Length p	Query result distribution		
			$q_{\min}$	$q_{\max}$	90% quantile
StateFlu	51	492	0	11,452	924
TDrive	100	672	0	39,871	1,772
Retail	1,298	374	0	372,306	15,089
TaxiPorto	1,298	672	0	317	2
WorldCup	1,298	1,320	0	16,928	0

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.

5.4 Analysis-Task-Specific Utility

Finally, we examine mechanism utility in an analysis task, namely anomaly detection, based on the framework in [43]. Specifically, we evaluate three anomaly detection techniques proven to be robust *and* effective [34] (LOF [5], KNN [32], and DWT-MLEAD [35]) on one real-world and two artificial one-dimensional streams.

We select the streams as follows: The real-world streams used in Section 5.2 and 5.3 do not feature anomaly labels. Hence, we select a stream with similar shape as the one-dimensional streams used in Section 5.2 from the streams of a recently proposed anomaly detection benchmark [43], namely *Dodgers*. It counts the number of cars next to a baseball stadium in 5 min. intervals. To be in line with the streams used in Section 5.2, we temporally aggregated the stream to 15 min. intervals. For the artificial streams, we extended our data generator by a post-processing step to place anomalies. Based on the stream properties amplitude and seasonality, we consider two streams featuring two different anomalies as follows: First, the *Pattern Anomalies* stream contains changes of the seasonal structure by extending the downtime between two consecutive seasons, i.e. a pattern shift. To this end, the generator randomly selects one season, finds the minimum $Q(D_t)$ between this and the next season (named $Q(D_t*)$), and inserts q additional timestamps with $Q(D_t*)+\mathcal{G}(0,1)$ before timestamp t^*. Second, the *Extrema Anomalies* stream contains changes of the amplitude by injecting extreme values. To this end, the generator selects v timestamps uniformly and modifies the respective $Q(D_t)$ to an unreasonably high value of $1.5 \cdot q_{max}$. The data generator input values for a, p, s, are the same as in Figure 1. Moreover, we use $q = \frac{s}{4}$ and $v = 11$. As in [34], we quantify the utility with the AUC-ROC score. For brevity, we depict the utility only for the anomaly detection technique performing best on the true stream.



(a) Results of vary-w experiments (ϵ = 1.0).

(b) Results of vary-ϵ experiments (w = 2).

Figure 6: Average AUC-ROC of vary-w and vary-ϵ experiments. ○ marks the baselines. AUC-ROC on the true streams: Dodgers 0.58, Pattern 0.73, and Extrema 1.0.

Figure 6 shows the average AUC-ROC score over 100 runs for all mechanisms and streams. Overall, we observe the expected result pattern (i.e., AUC-ROC score rapidly decreasing towards 0.5) for increasing w, and a slightly increasing trend when increasing ϵ. We identify three core observations: First, the window size w appears to be the decisive parameter for the utility. Second, even for small window sizes starting at w = 8, the AUC-ROC score has almost converged towards 0.5. Hence, all detection approaches have no detection quality beyond this value. Consequently, we selected $w = 2$ for the vary-ϵ experiment. This is the largest value where one clearly observes the positive effect of ϵ. Third, the utility differences between the mechanisms are rather small. Nevertheless, the baseline supremacy is still observable as either mechanism Sample or Uniform are among the best approaches.

To sum up, we observe that for relatively small window sizes $w \geq 8$, even the best anomaly detection techniques have no prediction quality. Therefore, we strongly recommend the inclusion of analysis tasks to quantify the significance of abstract utility improvements of novel mechanisms.

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

We provide recommendations that aim at controlling the expected utility of w-event DP mechanisms for data owners and administrators with expertise in data analysis rather than differential privacy.

Meaningful Privacy Requirements. Data owners are responsible for selecting the privacy requirements. For privacy budget ϵ, we recommend to select a value in $[0.1, 1.0]$. The default value of 1.0 should also be used in vary-w experiments, such that the expected noise per timestamp of baseline Uniform is w. For w, we cannot give data independent recommendations, but observe unnaturally large values. If the length of the longest event-sequence one intends to hide is not known, the length of a season s may serve as an upper bound for w. Our results furthermore suggest that analysis-task-specific metrics may indicate the analysis utility is poor for relatively small values of w. For instance, even the most robust and effective anomaly detection techniques from [34] cannot distinguish anomalies from normal timestamps for $w \geq 8$. This holds independent of the mechanism. To this end, a mechanism should not solely be selected by abstract error metrics, like MAE.

Consider the Selection of Baselines. Our study indicates to use Uniform if an expected error of $\frac{w}{\epsilon}$ is sufficient and query results are required for each timestamp, e.g., for *instant* change detection. Otherwise, time can be traded to minimize the perturbation error using a Uniform-Sample hybrid mechanism. This mechanism samples k times per window; 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 DP mechanism framework (cf. Algorithm 1).

isSamplingPoint-Function. In case the mechanism does not sample, the current query result is approximated with the last sanitized query result. This works well between seasons when the counts remain stable. However, it yields high errors in a growing or shrinking phase. Consequently, we suggest 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 DP 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 the best utility. Thus, mechanisms may limit the number of sampling timestamps in the current window.

Perturbation-Function. Our recommendation concerns 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 no longer correlated dimensions. Further, we question whether researchers should focus on dimension-grouping in future work since it violates privacy in case that the correlation of the dimension query results are not spurious. Otherwise, correlated dimensions result from an event that the data owner intends to hide. This may affect multiple rows in a database D_t and not only a single one as presumed in the original definition of DP [12]. The extension of DP with group-differential privacy [15] 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.

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 comparable benchmark that meets all requirements and conducted 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. Finally, we gave advise on mechanism selection and presented promising research directions in that field.

In future work, we aim to extend our micro benchmark to reveal novel insights on a mechanism's ability to improve data utility by exploiting spurious correlations in multi-dimensional streams. Further, we plan to investigate queries with sensitivity $\Delta Q > 1$, e.g., sum queries. We expect a heterogeneous influence on different functions of the mechanism framework.

REFERENCES

[1] Gergely Ács and Claude Castelluccia. 2011. I have a dream!(differentially private smart metering). In *International Workshop on Information Hiding*. Springer, 118–132.

[2] 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.

[3] 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.

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

[5] Markus Breunig, Hans-Peter Kriegel, Raymond Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. ACM, 93–104.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. 2019. Differential privacy in practice: Expose your epsilons! *Journal of Privacy and Confidentiality* 9, 2 (2019).

[14] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. 2010. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of Computing (STOC)*. 715–724.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

[23] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM J. Comput.* 40, 3 (2011), 793–826.

[24] Shiva P Kasiviswanathan and Adam Smith. 2014. On the'semantics' of differential privacy: A bayesian formulation. *Journal of Privacy and Confidentiality* 6, 1 (2014).

[25] 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.

[26] Jaewoo Lee and Chris Clifton. 2011. How much is enough? choosing ε for differential privacy. In *International Conference on Information Security*. Springer, 325–340.

[27] 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 on Conference on Information and Knowledge Management (CIKM)*. ACM, 1001–1010.

[28] 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.

[29] 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.

[30] Alisa Pankova and Peeter Laud. 2022. Interpreting Epsilon of Differential Privacy in Terms of Advantage in Guessing or Approximating Sensitive Attributes. In *2022 IEEE 35th Computer Security Foundations Symposium (CSF)*. IEEE, 96–111.

[31] 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.

[32] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient Algorithms for Mining Outliers from Large Data Sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. ACM, 427–438.

[33] 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.

[34] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly Detection in Time Series: A Comprehensive Evaluation. *Proceedings of the VLDB Endowment (PVLDB)* 15, 9 (2022), 1779–1797.

[35] Markus Thill, Wolfgang Konen, and Thomas Bäck. 2017. Time series anomaly detection with discrete wavelet transforms and maximum likelihood estimation. In *Proceedings of the International Conference on Time Series (ITISE)*. 11–23.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] Phillip Wenig, Sebastian Schmidl, and Thorsten Papenbrock. 2022. TimeEval: A Benchmarking Toolkit for Time Series Anomaly Detection Algorithms. *Proceedings of the VLDB Endowment (PVLDB)* 15, 12 (2022), 3678–3681.

[44] 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.