DPI: Ensuring Strict *D*ifferential *P*rivacy for *I*nfinite Data Streaming

Shuya Feng[†*], Meisam Mohammady[‡*], Han Wang[§], Xiaochen Li[¶], Zhan Qin[¶], Yuan Hong[†✉]
[†]*University of Connecticut,* [‡]*Iowa State University,* [§]*University of Kansas,* [¶]*Zhejiang University*
[†]{*shuya.feng, yuan.hong*}*@uconn.edu,* [‡]*meisam@iastate.edu,* [§]*han.wang@ku.edu,* [¶]{*xiaochenli, qinzhan*}*@zju.edu.cn*
[*]*Equal Contribution (Co-First Authors)*

***Abstract*—Streaming data, crucial for applications like crowdsourcing analytics, behavior studies, and real-time monitoring, faces significant privacy risks due to the large and diverse data linked to individuals. In particular, recent efforts to release data streams, using the rigorous privacy notion of differential privacy (DP), have encountered issues with unbounded privacy leakage. This challenge limits their applicability to only a finite number of time slots ("finite data stream") or relaxation to protecting the events ("event or w-event DP") rather than all the records of users. A persistent challenge is managing the sensitivity of outputs to inputs in situations where users contribute many activities and data distributions evolve over time. In this paper, we present a novel technique for *D*ifferentially *P*rivate data streaming over *I*nfinite disclosure (DPI) that effectively bounds the total privacy leakage of each user in infinite data streams while enabling accurate data collection and analysis. Furthermore, we also maximize the accuracy of DPI via a novel boosting mechanism. Finally, extensive experiments across various streaming applications and real datasets (e.g., COVID-19, Network Traffic, and USDA Production), show that DPI maintains high utility for infinite data streams in diverse settings. Code for DPI is available at https://github.com/ShuyaFeng/DPI.**

1. Introduction

Most computer systems and applications continuously generate data streams for analyses. For instance, online media such as YouTube [1] or Instagram recommends videos to users based on their vast visiting records. Network monitoring [2] detects abnormal behavior and identifies anomalies while tracking the network traffic. Water management [3] makes timely decisions on purifying water per the real-time water quality data collected by sensors.

Existing solutions have not effectively addressed the privacy threats posed by streaming data, which is vulnerable due to its volume and diversity. Recent attempts to release data streams using the widely adopted privacy model, differential privacy (DP), have revealed persistent privacy leakage [4], [5], [6], [7], [8], [9]. In particular, Dwork et al. [4] (STOC'10) found that in streaming data, as the number of rounds increases, an unavoidable logarithmic increase in error for an advancing counter occurs. Similarly, a logarithmic growth in privacy leakage is observed when the error is bounded. This issue arises from the infinite accumulation of two key factors in DP mechanisms: sensitivity (the maximum impact of each user on the result) and privacy budget composition (additional leakage from sequential result releases).

In the past decade, several solutions have been proposed to address "event-level privacy on finite or infinite streams" [5], [6], [7], [8], [9], [10], [11], [12], or "user-level privacy on finite streams under certain settings" (e.g., partially disclosing some randomized results and applying interpolation [13], [14], [15], [16], [17]). However, they have limitations in practice. The first category struggles when users' contiguous events collectively reveal sensitive information. Event-level privacy safeguards individual location visits but fails to protect a user's path across successive, potentially indefinite timestamps, like a sequence of locations. The second category is less practical for continuously and indefinitely collected data streams, since it is unrealistic to predict when services like traffic reporting will cease.

Thus, designing user-level DP mechanisms for infinite streaming involves addressing five fundamental challenges.

- **Unboundedness**. The data size in the stream is continuously growing and unbounded, with each user's data potentially generated indefinitely, making it also unbounded in infinite data streams.
- **Dynamic Data Distribution**. The data distribution changes dynamically over time, requiring DP mechanisms to consistently offer accurate analysis and privacy guarantees upon these changes.
- **Accumultative Privacy Loss over Infinite Streams**. This has been widely recognized but not addressed yet.
- **Utility-Privacy Tradeoff**. It is challenging to minimize the utility loss continually (under a bounded leakage) while leveraging dynamic changes in data distribution.
- **Real-Time Processing**. DP mechanisms for infinite data streams must balance real-time processing needs with privacy protection and computational efficiency.

To our best knowledge, existing methods have not adequately addressed the significant challenges in infinite data streams. Challenges in current methods include not supporting infinite data disclosure, assuming limited sensitivity instead of indefinite full event sequence protection, and limiting applications to fixed time slot queries. Table 1

TABLE 1. REPRESENTATIVE DP DATA STREAMING METHODS (NOT SCALABLE TO USER-LEVEL PROTECTION OVER INFINITE DATA STREAMS). OTHER EXISTING METHODS SHARE SIMILAR PROPERTIES.

Methods	Assumption	Properties in Existing Works		
	(Sensitivity, Setting)	Privacy	Error Bound	Applications
[6]	(1, event)	$O(\epsilon \cdot t)$	$O(\frac{\log t^{1.5} \cdot \log \frac{1}{\delta}}{\epsilon})$	count, top-k, range queries
[7]	(1, event)	$O(\epsilon \cdot t)$	-	count, event monitoring
[9]	(1, event)	$O(\epsilon \cdot t)$	$O(\tau(\log t)^{1.5})$	count, range queries, etc.
[10]	(1, event)	$O(\epsilon \cdot t)$	w.r.t. query range, threshold, ϵ, t	multiple analyses

provides an overview of the most crucial properties of representative existing works on data streaming methods with differential privacy. There are many limitations in existing methods. For instance, they all consider the special case of sensitivity (e.g., 1 for counting) and their privacy bound linearly increases with the number of rounds t.

In this paper, we present a novel DP framework for infinite data streams, known as DPI (*Differential Privacy for Infinite Data Streams*). DPI is designed with three key components to address the above challenges: (1) *Sensitivity Compression*, (2) *DPI Boosting Mechanism*, and (3) *Privacy Budget Allocation Mechanism*, discussed as below.

1.1. DPI in a Glimpse

Sensitivity Compression. Bounding sensitivity is crucial for reliable DP models. DPI adopts a *probability distribution data structure* to effectively limit the unbounded sensitivity by projecting/compressing the maximum output difference to the ℓ_1-norm of probability distribution difference, offering advantages over unbounded structures like counting.

DPI Boosting Mechanism. The design of DPI has two inherent crucial challenges: (1) preserving a (lower-bound) meaningful utility for the data stream, and (2) developing methods to capture dynamic patterns within the new coming data. A sequence of randomization mechanisms (e.g., additive noise [18], sampling [19] or randomized response [20]) for all the time slots, inevitably destroys both cumulative and instantaneous information. For instance, applying a series of ϵ-DP Laplace mechanisms over a dataset D will generate results with an error of $O\left(2^n / \left(|D| \cdot \epsilon\right)\right)$ [21].

Unlike atomic DP mechanisms, additive or accumulative learning algorithms are especially effective for streaming data, where they continuously update models in real-time to adapt to evolving patterns. In particular, Dwork et al. [22] and Moritz et al. [23] demonstrated that the differential privacy version of boosting mechanisms can be effective in achieving cumulative learning outcomes for a finite number of queries. However, direct application of these approaches to continual streaming data results in *unbounded privacy leakage* and *significant computational overhead* [22], [23]. We have identified that both deficiencies result from the use of an expensive *base synopsis generator* in [22], [23].[1]

To handle infinite streams, we have devised a novel DPI boosting mechanism, which guarantees both bounded and efficient computation while integrating an infinite number of tiny-budget DP mechanisms into an additive learning game. While the outcomes of various queries on streaming data may vary over time, it is still possible to precisely associate each result with an adequately accurate state, such as a numerical value or a PDF. The DPI boosting mechanism introduces a groundbreaking approach to synopsis generation by creating a significantly smaller set of synopses from PDFs, resulting in faster processing. Such non-trivial synopsis generation is designed as a privacy-free (0-DP) mechanism, generating an ample number of synopses that precisely address a set of queries across an infinite number of rounds. It necessitates a rigorous analytical analysis to determine the number of synopses required to achieve a meaningful level of accuracy while ensuring privacy. Also, the existing uni-directional boosting mechanism [22], [23] (updating weights solely over queries based on data-dependent synopses) should be revised into a *bi-directional boosting* approach that considers both data-independent synopses and queries for infinite data streaming (*otherwise, the privacy budgets will not be sufficient*). With these considerations in DPI, we aim to achieve provable utility while maintaining the bounded privacy guarantees.

Privacy Budget Allocation and Boosting. By saving budget on the synopsis generation process in DPI, we develop a bounded budget boosting mechanism that optimally utilizes an infinite number of tiny boosting mechanisms for making refinements, focusing the privacy budgets solely on *reweighting* and not synopses. To achieve this, we establish an optimal allocation strategy with a converging sum series to maximize the utility.

To prevent budget depletion after a certain number of releases, typically expected with a decaying series, we employ a sophisticated deployment of this infinite decaying series of privacy budgets. During each iteration (time slot) of DPI, we randomly generate the privacy budget from the series, allowing for efficient and controlled management of the privacy budget throughout the continuous process.

Figure 1 shows an overview of DPI, including the input data stream, the three key components, and the output results (as streaming PDFs). The DPI framework takes an incoming data stream as input and processes it through sensitivity compression, privacy budget allocation, and a boosting mechanism to output an infinite series of differentially private results that maximize accuracy and utility over time (see the detailed design in Section 4). This end-to-end process allows DPI to ensure strong privacy for users while handling the challenges of infinite streaming data. Note that DPI may require a restart to refresh the privacy budget in possible uncommon cases of extremely long-term, highly dynamic data streams, as discussed in Section 7.

1.2. Contributions

To sum up, DPI makes the following major contributions:

Unique and Significant Benefits. First, to our best knowledge, DPI provides the first user-level DP solution for

1. Such generator acts as a compact representation of query answers over a dataset and can be implemented using various forms, such as a synthetic database or any data structure such as kernel or probability density function (PDF) that approximates query results ("synopsis") [22], [23].

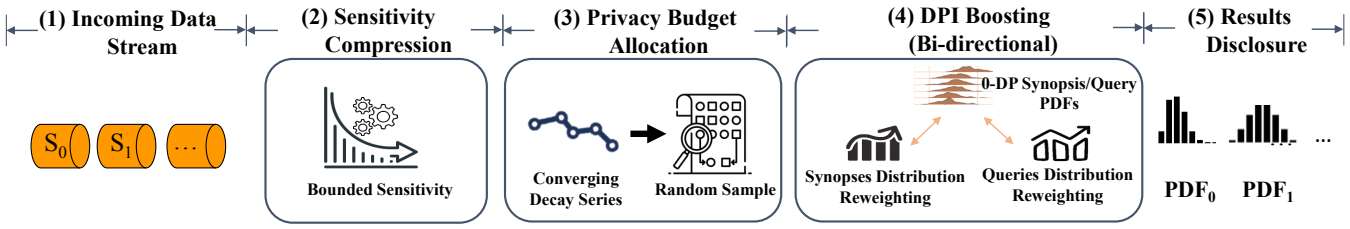


Figure 1. DPI guarantees limited privacy bound and sensitivity by managing the pace of convergence and error margins, striving to attain the highest level of accuracy (*the output streaming PDFs can support most real-time applications as the input stream*).

protecting users in applications relying on continuously collecting data over infinite streams, e.g., YouTube [1] or Instagram recommends videos to users based on their vast visiting records indefinitely. Second, unlike most existing works which usually take various limited assumptions (e.g., bounded sensitivity, protecting the presence/absence statuses of events rather than users, a limited number of disclosures), DPI strictly satisfies generic ϵ-DP for protecting all the users (end-to-end) involved in the unlimited data streams. Third, DPI provably maximizes the accuracy of the disclosure in the long run with substantial theoretical analyses.

Novel Methods. We have designed a novel dynamic boosting mechanism (DPI boosting) that optimally balances both privacy and utility over time. DPI significantly differs from the "AdaBoosting" [22] for DP: (1) DPI replaces the base synopsis generator with a pre-determined set of synopses *independent of the dataset*. We demonstrate that these synopses can be created by the performance of the output and strong synopses can be integrated into the boosting mechanism with no impact on privacy, and (2) DPI boosting has been designed to generate a PDF data structure that tracks the overall PDF of the stream instead of accurately answering pre-defined queries. We have also established lower bounds on the utility of DPI based on this PDF data structure.

Comprehensive Evaluations. We comprehensively evaluate the performance of DPI on different applications and datasets. Our primary objectives in these evaluations are: (1) investigating the impact of DPI's hyperparameters on its performance over time (in Section 6.2), (2) evaluating DPI on different datasets with different domain sizes/distributions and a wide variety of real applications, such as statistical queries, anomaly detection, recommender systems (in Sections 6.3 and 6.4), and (3) compare DPI with existing methods and show that they violate the privacy requirement ϵ after a few rounds (in Appendix E.1) while DPI strictly bounds the privacy leakage with ϵ indefinitely and still has considerable share of unspent privacy budget after a large number of rounds.

2. Preliminaries

2.1. Streaming Model

We consider data collection and analysis in the context of interactive and continuous observation over infinite data streams. In this setting, a trusted curator maintains a dynamic dataset, represented as a collection of streaming input $S = \{S_1, S_2, \ldots\}$, where S_t refers to the indexed input collected at time slot t. The universe of possible individual records is denoted as X.

Interactive Queries. The data analyst interacts with the curator by submitting a sequence of queries over time. Let Q be the space of all possible queries, and at each time slot t, the data analyst requests a subset of queries $Q_t \subseteq Q$. Each query q_t is a function that takes the data until the current stream S_t and outputs a response $y_t = q_t(S_1, S_2, \ldots S_t)$, which is released by the curator. Note that q can remain the same as previous time slots or dynamically change.

2.2. Privacy Models

To protect the privacy of individual records in the context of continual observation, the trusted curator employs Differential Privacy (DP) mechanisms represented by $\mathcal{M}(\epsilon)$, where $\epsilon > 0$ is the privacy budget allocated for each query in the infinite data stream. Each query q_t outputs a response $y_t = q_t(S_1, S_2, \ldots, S_t)$, and the curator releases a private response denoted as $\tilde{y_t}$. The DP mechanism $\mathcal{M}(\epsilon_t)$ ensures that $\tilde{y_t}$ does not disclose any information about the presence or absence of sensitive information from any particular individual in all the streamed data. By limiting the privacy budget ϵ_t, the curator controls the amount of privacy provided for each query q_t.

We adopt the standard definition of ϵ-DP [18]. Let Γ be the domain of datasets, and let D, D' be two datasets in Γ such that D' can be obtained from D by adding or removing the data of a single individual (user). A randomization mechanism $\mathcal{M} : \Gamma \times \Omega \to \mathbb{R}$ is ϵ-differentially private if, for all $\Theta \subseteq \mathbb{R}$ and for all $D, D' \in \Gamma$ such that D' is adjacent to D, i.e., D and D' differ in the data of a single individual, the following inequality holds:

$$\mathbb{P}(\mathcal{M}(D) \in \Theta) \leq e^{\epsilon}\mathbb{P}(\mathcal{M}(D') \in \Theta) \tag{1}$$

where Ω is a sample space by randomization $\mathcal{M}$ to generate the output. To achieve differential privacy, we need to consider the concept of sensitivity. The sensitivity Δ [24] of a query q is defined as the maximum ℓ_1-distance between the exact query answers on any two neighboring databases D and D', i.e., $\Delta q = \max_{D,D'} \|q(D) - q(D')\|_1$. Bounding sensitivity is crucial for DP models. Even with a constrained

privacy budget, unbounded sensitivity can result in overly sensitive outputs, risking privacy breaches.

2.3. Boosting Mechanisms

We next introduce some background for the AdaBoosting mechanism and DP with AdaBoosting, which will be significantly revised on both *privacy* and *efficiency* for DPI.

2.3.1. AdaBoosting Mechanism. AdaBoosting (Adaptive Boosting) [22] is a powerful machine learning technique used to enhance the accuracy of weak classifiers by combining them into a strong ensemble classifier. Formally, given a training dataset $D = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where x_i and y_i represent the data points and their respective labels, AdaBoosting constructs an ensemble classifier $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$. Here, T is the number of weak classifiers, $h_t(x)$ is the t-th weak classifier, and α_t is the corresponding weight assigned to $h_t(x)$. The objective of boosting is to convert a weak learner, which produces a hypothesis only slightly better than random guessing, into a strong learner that achieves high accuracy. The AdaBoosting proceeds in rounds. In each round, two steps are performed:

1) The base learner is run on the current distribution $\mathcal{D}_t$, generating a classification hypothesis h_t.
2) The hypotheses $h_1, h_2, \ldots, h_t$ are used to reweight the samples and define the updated distribution $\mathcal{D}_{t+1}$.

It continues for a predefined number of rounds or until the combination of hypotheses is sufficiently accurate.

2.3.2. DP with AdaBoosting. DP with AdaBoosting, introduced by Dwork and Rothblum [22], combines AdaBoosting with differential privacy. As detailed in Figure 12 (in Appendix A), the process iteratively updates the weights assigned to each query, and samples queries based on their weights as follows. Let $Q = \{q_1, q_2, \ldots, q_m\}$ represent a set of queries. The method assigns uniform weights w_i to each query q in Q in its initialization phase ($P_Q(t = 1) \rightarrow$ uniform). Next, it will operate on the "base synopsis generator" to produce "weak" DP synopses $\mathcal{A}^t_{Q_s}$, which are moderately accurate for r sampled queries Q_s.

Definition 1 (Sampled Queries Q_s)**.** *Let Q be the set of requested queries, and $P_Q(t)$ be the PDF for the query sampling at iteration t in DP AdaBoosting [22]. At each iteration t, the set of sampled queries Q^t_s is defined as a k-size subset of Q, sampled from Q per $P_Q(t)$. The sampling process reflects AdaBoosting's practice of assigning larger weights to weaker queries to improve their performance.*

Definition 2 (Synopses)**.** *A synopsis [25], [22] generated from a dataset D is defined as a compact representation of answers to a set of queries $Q = \{q_1, q_2, \cdots, q_k\}$ over D. Synopsis can be in the form of a synthetic database, or any arbitrary data structure, e.g., a kernel or PDF, that can be queried to return an approximation of the query's result.*

This data structure is referred to as a *synopsis* of the database. General privacy-preserving synopses are of interest as they may be easier to construct than privacy-preserving synthetic databases. Also, stronger limitations are known for constructing synthetic databases than general privacy-preserving synopses.

Definition 3 (Base Synopsis Generator [22])**.** *For any data universe X and set of queries $Q : \{X^n \rightarrow \mathbb{R}\}$ with sensitivity Δ_Q, a $(r, \lambda, \varepsilon_1, \kappa)$-base synopsis generator for Q is a method that:*

- *Outputs a synthetic database y of size n based on $r = O(n \cdot \log|X| \cdot \kappa)$ queries from Q, where κ is the probability that the chosen synopsis is λ-accurate (the synopsis closely approximates the true data within a λ margin of error).*
- *Adds Laplace noise $O(\kappa\sqrt{r} \cdot \frac{\Delta_Q}{\varepsilon_1})$ to each query answer.*
- *Guarantees $(\varepsilon_1, \exp(-\kappa))$-differential privacy.*
- *It has a complexity of $|X|^n \cdot poly(n, \kappa, \log(1/\varepsilon_1))$.*

The base synopsis generator outputs the lexicographically first database y satisfying $|q(y) - \hat{q}^t_s(x)| \leq \lambda/2$ for every query q in the sampled query set Q_s, where $\hat{q}^t_s$ is the noisy query answers. If no such database exists, it outputs $\perp$ (i.e., fails). The $(r, \lambda, \varepsilon_1, \kappa)$-base synopsis generator privately generates a synthetic database approximating the responses of the original database to a set of queries Q.

The goal is to iteratively improve the base generator using the boosting algorithm to create a synopsis that performs well for all requested queries Q while preserving privacy.

The next steps aim to strengthen the queries with bad results by adaptive sampling and refining them. At each iteration, the query sampling probability $P_Q(t)$ is reweighted based on the performance of the base synopsis generator over all queries such that the probability modification is to iteratively assign higher weights to the (new) queries with bad results, allowing the subsequent weak model to focus on these challenging examples and improve overall accuracy. Finally, it integrates a learning rate η^Q_t in the reweighting process, which is proportional to the privacy budget allocated for boosting (ϵ_2). Then, two separate privacy budgets are required: one for generating synopses (ϵ_1) and the other for the reweighting step (ϵ_2). Consequently, AdaBoosting with DP [22] becomes a privacy-expensive mechanism.

DP with AdaBoosting's runtime depends on the number of queries and the base synopsis generator's runtime. The accuracy of the mechanism scales with $\sqrt{n}\log m$ (n: size of the data, m: number of queries). For details of the synopses and base synopsis generator, see [25], [22].

3. Overview of DPI Framework

In this section, we present the DPI framework comprising the DPI Interface and the DPI Brain. The DPI Interface manages temporal input streams, stores sequential DP results for pre-defined queries ("query synopses"), and delivers query results to analysts over time. On the other hand, the DPI Brain focuses on online learning. The details of the DPI framework are shown in Figure 2.

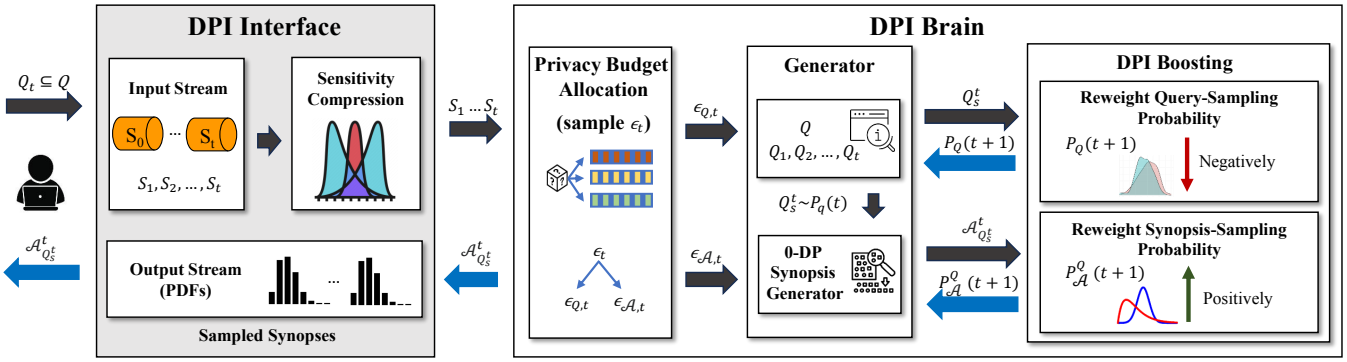


Figure 2. The DPI framework in time slot t. (1) Q_s^t can be a single query or a subset of queries, and can remain the same or dynamically change over time, (2) all the privacy budgets are pre-computed with converging series and ϵ_t is sampled in real-time, and halved for query-sampling and synopsis-sampling, (3) generator returns the PDF output (sampled synopsis) for the current time slot t to the interface/analyst, and also sends it for boosting, and (4) DPI boosting (bi-directional) reweights the PDFs (both query-sampling probability and synopsis-sampling probability) for the next time slot $(t+1)$.

3.1. DPI Components

DPI Interface plays a central role in managing the interaction between analysts and the underlying data, including:

- *Input Module* collects temporal data streams from users over successive timestamps. It also receives temporal queries $Q_1, \ldots, Q_t$ from the analyst.
- *Sensitivity Compression* processes the input data stream with a PDF data structure to bound sensitivity.
- *Output Module* returns temporal output query synopses as a sequence of PDFs with DP guarantees for a pool of queries $Q_1, \ldots, Q_t$, which can be converted to support most data analyses on the input stream.

DPI Brain comprises the following modules.

- *Privacy Budget Allocation* generates the privacy budget ϵ_t for each time slot t.
- *Generator Module* generates the sampled queries (Q_s^t) and a set of sampled synopses $(\mathcal{A}_{Q_s^t}^t)$ for each query $q \in Q_s^t$. It interacts with the DPI Boosting module, receiving the updated configurations in terms of PDFs to effectively adjust the sampling process.
- *DPI Boosting Module (bi-directional)* effectively captures the dynamic nature of data streams (new coming data and queries) when answering queries.

3.2. DPI Workflow

- **Step 1:** *Input Streaming and Sensitivity Compression.* See details in Section 4.1.
- **Step 2:** *Privacy Budget Allocation.* In time slot t, DPI randomly samples a pre-computed privacy budget ϵ_t from the series, halved for query-sampling and synopsis-sampling (see details in Section 4.4).
- **Step 3:** *0-DP Synopsis Generation.* The Generator module generates a set of parameters, including a sampled subset of queries Q_s^t over the pre-defined pool of queries Q and a set of sampled synopses $\mathcal{A}_{Q_s^t}^t$ for queries $q \in Q_s^t$. To derive the sampled synopses $\mathcal{A}_{Q_s^t}^t$, we need to generate a pool of synopses $\mathcal{A}$ that can represent all possible distributions in a dataset. This generation can avoid consuming privacy budget (0-DP, see details in Section 4.2).
- **Step 4:** *DPI Boosting.* The DPI Boosting module improves accuracy via bi-directional reweighting of queries and synopses (see details in Section 4.3).
  - *Reweighting Synopsis-Sampling Probability.* In this phase, the DPI Boosting module uses a portion of the privacy budget allocated for DPI to perform reweighting of the Synopsis-Sampling Probability. This step aims to improve the accuracy of synopses and update the sampling PDF accordingly.
  - *Reweighting Query-Sampling Probability.* In this phase, another portion of the privacy budget is used to reweight the Query-Sampling Probability. This step ensures that the most informative queries are given higher probabilities in the sampling process.
- **Step 5:** *Output Private Streams (PDFs).* DP query results (temporal) are provided to analysts over time.

4. Detailed Design in DPI

4.1. Sensitivity Compression

To enable effective privacy for infinite data streams, the sensitivity of the analysis must be bounded. Sensitivity represents the maximum change in the output resulting from the presence or absence of any single user's data. This motivates the need for sensitivity compression in the infinite stream setting. Our key insight is adopting a PDF data structure to represent the data stream. This provides an inherent sensitivity bound, as changing one user's data can only alter the PDF by a maximum total variation distance of 2 (considering the two PDFs of two adjacent inputs are two normalized vectors with all the entries summing up to 1 in each). Denote D and D' as the output distributions of two neighboring inputs for any query at time slot t. Thus, for all D and

D', their sensitivity is derived as $\Delta_Q = ||D - D'||_1 \leq 2$. By bounding the PDF difference, we effectively compress sensitivity to a fixed constant for all the queries. This allows formal privacy guarantees to be achieved over infinite data. Furthermore, the PDF representation maintains high utility for the analyses in most real-time applications.

4.2. 0-DP Synopsis Generator

The synopsis generation in DPI is responsible for generating a sampled synopsis for each query $q \in Q_s$. Our key observation is that establishing a universal synopses pool can be done in a privacy-free (0-DP) and efficient way, enabling the system to privately answer a wide range of queries.

Establishing a Pool of PDFs. DPI utilizes a pioneering 0-DP synopsis generation process to navigate this challenge (AdaBoosting [22] cannot solve). This method constructs a set of PDFs that can universally emulate any PDF with minimal error (see detailed analysis in Appendix C). These PDFs are then employed to generate answers for a set of queries, delivering approximated results without requiring access to raw data. Various types of queries can be addressed using these PDFs, including but not limited to Point Queries, Range Queries, Aggregation Queries, Top-K Queries, Join Queries, and Correlation Queries [26], [27], [28], [29].

Quantization. The first step in the 0-DP synopsis generation process is quantization, which discretizes the values that PDFs can take. We define the precision level of quantization as p, representing one unit of probability (e.g., $p = 0.001$). By dividing the range of possible values into intervals of size p, we can express the quantized value of a continuous variable x as $\text{quantize}(x) = \text{round}(x/p) \cdot p$. This ensures that the values are discretized into intervals of width p.

Sampling Synopses with Multinomial Distribution. After quantization, we can generate all possible PDFs by distributing the available units of probability (denoted as $n_p = \frac{1}{p}+1$) among the domain of interest, denoted as $\mathcal{K}$. Since n_p should be the same as the dataset size n, we can derive the value of p. Then, we can model this distribution using a multinomial PDF, denoted as $P(x)$, where x represents the random variable that takes on values from the domain $\mathcal{K}$. The multinomial PDF captures the probabilities of outcomes for this discrete random variable with k categories.

During each multinomial sampling, we treat each category equally and sample them uniformly for n_p times, leading to a generation of a synopsis (the distribution of n data points with k categories). To create the 0-DP synopses pool, we leverage the multinomial sampling. Let n be the number of samples drawn from the multinomial distribution and denote the sampled values as $(x_1, x_2, \ldots, x_n)$, where each x_i represents a possible outcome from the multinomial distribution defined by $P(x)$. We should repeat the multinomial sampling process for N times to get the pool of synopses $(P_1, P_2, \ldots, P_N)$. The algorithm of 0-DP synopsis generator is shown in Appendix C.

Next, for queries in Q_s^t, a set of synopses $\mathcal{A}_{Q_s^t}^t$ should be sampled according to the most recently updated sampling probability $P_{\mathcal{A}}(t)$. We will discuss how to update the sampling weights for different synopses in the DPI Boosting. The DPI Boosting algorithm iteratively improves the accuracy of DP queries on streaming data through a bi-directional reweighting process that favors challenging queries and high-quality synopses. By continually reweighting query and synopsis sampling probabilities based on estimated errors, DPI Boosting adapts the new coming data to optimize accuracy under DP constraints.

0-DP Guarantee for the Synopsis Generator. The synopsis generation process in DPI is uniquely designed to ensure that it does not consume any privacy budget since it operates without accessing to any raw data. Initially, the Synosis Generator invokes quantization to discretize potential values for any PDFs without using any raw data. Subsequently, through the procedure of multinomial synopses sampling, all potential PDFs are generated by distributing available units of probabilities across a defined domain of interest, bypassing the need for raw data access. The detailed steps and theoretical analysis are given in Appendix C.

4.3. DPI Boosting (Bi-directional)

The DPI Boosting continuously evaluates and adjusts the importance of a subset of queries ("sampled queries"). It allows less accurate sampled queries to be reweighted higher, while the reweighting process is applied to focus more on accurate synopses for each sampled query. This bi-directional boosting process effectively captures the dynamic nature of data streams when answering queries. In this section, we will provide a comprehensive understanding of the DPI Boosting module and its role in enhancing the utility and privacy guarantees of the DPI.

Bi-directional Boosting. DPI Boosting significantly revises the AdaBoosting under DP constraints [22] to reweight the sampling weights. Specifically, we focus on the optimization techniques used to reweight the query and synopsis distribution during each round of DPI Boosting. The goal is to assign higher weights to poorly handled queries, and lower weights to well-handled queries, ultimately leading to enhanced accuracy and utility of the final synopsis.

Recall that we sample a subset of synopses $\mathcal{A}_{Q_s^t}^t$ for the sampled subset of queries Q_s^t. In the DPI Boosting, we should evaluate the difference between the true query result and the query result based on the synopsis. Our weight boosting differs from the AdaBoosting [22] from two aspects: (1) there is a subset of synopses instead of one synopsis to reweight the sampling probability, and (2) the reweighted sampling probability should be used both for queries in a pool and synopses in a pool. The details of DPI boosting are shown in Algorithm 1. Notice that the size of query pool Q may be different from the size of synopses pool $\mathcal{A}$ and it should not affect the reweighting process since normalization is applied for queries and synopses.

Choice of Parameters in DPI Boosting. The reweighting depends on the parameters λ, μ, and η, which all evaluate the accuracy of synopses. Note that λ denotes the error bound

Algorithm 1: DPI Algorithm

Input: data stream S, privacy budget ϵ, synopsis error bound λ, error bound for overfitting μ, decay rate of decaying series ζ, query pool Q, synopsis pool $\mathcal{A}$ (Algorithm 2)
Output: private output streams $\mathcal{A}^t_{Q^t_s}$ (PDFs)

Initialize $P_Q(1)$, $P_{\mathcal{A}}$ to uniform distribution
foreach S_t *in Stream* S **do**
  /* Random Budget Allocation */
  $\epsilon_t = \text{RBA}(t)$ /* Algorithm 3 */
  $\epsilon_{Q,t} = \epsilon_{\mathcal{A},t} = \frac{\epsilon_t}{2}$
  /* Q^t_s sampling */
  $Q^t_s \sim P_Q(t)$
  /* $\mathcal{A}^t_{Q^t_s}$ sampling */
  $\mathcal{A}^t_{Q^t_s} \sim P_{\mathcal{A}}(t)$
  /* DPI Boosting */
  if $|q(S_t) - \mathcal{A}^t_{Q^t_s}| < \lambda$ **then**
    $P_{\mathcal{A}}(t)[q] \leftarrow 1$, $P_Q(t) \leftarrow -1$
  if $|q(S_t) - \mathcal{A}^t_{Q^t_s}| \geq \lambda + \mu$ **then**
    $P_{\mathcal{A}}(t)[q] \leftarrow -1$, $P_Q(t) \leftarrow 1$
  else
    $P_{\mathcal{A}}(t)[q], P_Q(t) \leftarrow 1 - 2(|q(D) - \mathcal{A}^t_{Q^t_s}| - \lambda)/\mu$
  $P_{\mathcal{A}}(t+1)$ = Normalize $P_{\mathcal{A}}(t)$
$u_{q,a} \leftarrow exp(-\alpha_t \cdot \sum_{j=1}^{t} P_{\mathcal{A}}(t)[q])$, where $\alpha_t = \frac{1}{2}\ln[(1+2\eta_t)/(1-2\eta_t)]$ and η_t is the learning rate in time slot t
normalization factor $Z_t \leftarrow \sum_{q \in Q} u_{q,a}$
update $P_Q(t+1) = u_{q,a}/Z_t$, $P_{\mathcal{A}}(t)[q] = u_{q,a}/Z_t$
return $\mathcal{A}^t_{Q^t_s}$ as private output streams

of the chosen synopses, while μ refers to an additional error bound that is tolerated to avoid overfitting during DPI boosting. The parameter η is determined by the specified privacy budget ϵ. We find the optimal value of η_t for each time slot with the given ϵ to minimize the error bound of DPI framework under the privacy constraints. The details are presented in Section 5.2 (Theorem 5.4).

4.4. Random Budget Allocation (RBA)

As outlined in Algorithm 3 (in Appendix D) to mitigate the risk of *systematic depletion* over time, DPI employs a random selection process when defining ϵ_t ("Random Budget Allocation (RBA)") rather than sequentially picking elements from the infinite series of potential budget values. It employs an exponential PDF with $p(\epsilon_t = x) = \exp(-\Lambda \cdot x)$, where Λ is an astronomically large number. This could generate very tiny samples, with the probability of obtaining such small values approaching 1.

After generating such very tiny values, they are mapped to the elements in the series ε sampled from the exponential PDF. The closest one to the sampled value is then returned, denoted as $\epsilon_t \leftarrow \arg\min_{\epsilon \in \varepsilon_t} |\epsilon - S|$. We note that since RBA generates samples without replacement in the series, they tend to get smaller and smaller over time. However, the sum of t independent samples from an exponential PDF with the rate parameter Λ follows a Gamma distribution with a shape parameter t and a rate parameter Λ. The mean (υ) of a Gamma distribution with shape parameter ω and rate parameter θ is given by $\frac{\omega}{\theta}$. In our case, for the sum of t exponential samples, given the shape parameter t and the rate parameter Λ, a mean (υ) of $\frac{t}{\Lambda}$ can be generated. Therefore, for any $t < \Lambda$, we can effectively preserve $\epsilon - \frac{t}{\Lambda}$. In practice, this bound must be much better as samples of the series are tending to be smaller than exponential PDF's samples due to sampling without replacement. An alternative RBA scheme based on *ranges* is given in Appendix D.

Finally, as shown in Figure 2, each ϵ_t is further equally divided into $\epsilon_{Q,t}$ and $\epsilon_{\mathcal{A},t}$ for reweighting the query and synopses distribution, respectively. Specifically, it determines the learning rate that controls how aggressively the query sampling distribution is updated based on the empirical errors. Also, $\epsilon_{\mathcal{A},t}$ plays a key role in governing the privacy budget utilized when reweighting the synopses distribution $P_{\mathcal{A}}(t)$. By separating the budgets for querying and synopses sampling, DPI allows more flexible control to boost each of these two key components in the overall mechanism. Figure 10 plots the remaining budgets after t time slots and it clearly demonstrates that RBA is a thoughtful budget consumption strategy over the infinite disclosure.

DPI Framework. Algorithm 1 presents the details for DPI, which outputs PDF(s) in each time slot to dynamic queries with ϵ-DP over infinite streams. W.l.o.g., we take the single-query case for Q^t_s as an example. A set of queries for Q^t_s only need to split the budget for multiple output PDFs.

5. Privacy and Utility Analysis

5.1. Privacy Loss after t Iterations

To understand the privacy guarantees of our DPI, we analyze the privacy budget consumed over successive disclosures. Theorem 5.1 shows how to calculate the total privacy budget ϵ by DPI after t iterations (e.g., time slots).

Theorem 5.1. *Consider a series of DPI mechanisms, each triggered with η^Q_i and $\eta^{\mathcal{A}}_i$. The overall privacy budget of DPI after t iterations is given by:*

$$\epsilon = \frac{4}{\mu} \sum_{i=1}^{t} \log \left[\left(\frac{1+2\eta^{\mathcal{A}}_i}{1-2\eta^{\mathcal{A}}_i} \right) \left(\frac{1+2\eta^{Q}_i}{1-2\eta^{Q}_i} \right) \right] \tag{2}$$

Proof. Each round of disclosure t results in the use of an additional privacy budget $D_\infty(\mathcal{D}_t || \mathcal{D}'_t)$. Dwork et al. [22] showed that, under the assumption that synopses are known, this quantity is $\frac{2}{\mu}\log\left(\frac{1+2\eta_t}{1-2\eta_t}\right)$ privacy budget. Briefly, for each query $q \in Q$, let $d^t_q = |q(x) - A_t(q)|$ and $d'^t_q = |q(x') - A_t(q)|$. It follows that $|d^t_q - d'^t_q| \leq \Delta_Q$. This implies $|P_{\mathcal{A}}(t)[q] - P'_{\mathcal{A}}(t)[q]| \leq \frac{2\Delta_Q}{\mu}$, and thus:

$$e^{-2\alpha \cdot t \cdot \frac{\Delta_Q}{\mu}} \leq \frac{u_t, q}{u'_t, q} \leq e^{2\alpha \cdot t \cdot \frac{\Delta_Q}{\mu}}$$

Define normalization factor for two adjacent database as $Z_t = \sum_{q \in Q} u_t, q$ and $Z'_t = \sum_{q \in Q} u'_t, q$, we have:

$$e^{-2\alpha \cdot t \cdot \frac{\Delta_Q}{\mu}} \leq \frac{Z_t}{Z'_t} \leq e^{2\alpha \cdot t \cdot \frac{\Delta_Q}{\mu}}$$

The claim follows from the above because $\mathcal{D}_{t+1}[q] = \frac{u_t,q}{Z_t}$, $\mathcal{D}'_{t+1}[q] = \frac{u'_t,q}{Z'_t}$ and replacing $\alpha = \frac{1}{2}\log\left(\frac{1+2\eta}{1-2\eta}\right)$.

Therefore, in DPI, since we update two PDFs $P_Q(t)$ and $P_{\mathcal{A}}(t,q)$, each of these two PDFs will consume a privacy budget of $\frac{4}{\mu}\sum_{i=1}^{t}\log\left(\frac{1+2\eta_i^{\mathcal{A}}}{1-2\eta_i^{\mathcal{A}}}\right)$ and $\frac{4}{\mu}\sum_{i=1}^{t}\log\left(\frac{1+2\eta_i^{Q}}{1-2\eta_i^{Q}}\right)$, respectively. This completes the proof. □

The convergence of the privacy bound in the DPI can be analyzed through the total privacy budget ϵ utilized after t iterations. Next, assuming the privacy budet is divided equally between the two reweighting mechanisms, i.e., $\epsilon_Q = \epsilon_{\mathcal{A}} = \frac{\epsilon}{2}$, the optimal series can be derived.

5.2. Total Privacy and Utility Loss

According to Dwork et al. [22] (Claim 4.3), after t rounds of DP with AdaBoosting, while the mechanism produces λ-inaccurate results, the loss probability $\mathcal{L}(t)$, derived via the cardinality of a subset of queries ($Q_{\text{bad}} \subset Q$), is upper-bounded by $\left(\sqrt{1-4\eta^2}\cdot t\right)\cdot|Q|$. A very similar observation to the proof but utilizing different η_i in each iteration i (from a converging series) will lead to $\sum_{i=1}^{t}\mathcal{L}_i = \sum_{i=1}^{t}\frac{1}{2}\log\left[(1+2\eta_i)(1-2\eta_i)\right]$.

Theorem 5.2 (Proof in Appendix B.1)**.** *Consider a series of DPI mechanisms, and each triggered with η_i^Q and $\eta_i^{\mathcal{A}}$. The overall error of DPI, represented as $\sum_{i=1}^{t}\mathcal{L}_i = [\mathbb{P}(|\mathcal{A}(q,t) - q(S_1,S_2,\cdots,S_t)| > \lambda+\mu)]$, is upper-bounded by:*

$$\frac{1}{4}\sum_{i=1}^{\infty}\log\left[(1-4(\eta_i^Q)^2)\cdot\left(1-4(\eta_i^{\mathcal{A}})^2\right)\right] \tag{3}$$

Theorems 5.1 and 5.2 allow us to formulate an optimization problem for calculating the parameter η to maximize utility under a total privacy constraint.

$$\min_{\eta_i, i\in[\mathbb{N}]}\ \frac{1}{4}\sum_{i=1}^{N\to\infty}\log\left[(1-4(\eta_i^Q)^2)\left(1-4(\eta_i^{\mathcal{A}})^2\right)\right]$$
$$\text{w.r.t. } \frac{4}{\mu}\sum_{i=1}^{N\to\infty}\log\left[\left(\frac{1+2\eta_i^{\mathcal{A}}}{1-2\eta_i^{\mathcal{A}}}\right)\left(\frac{1+2\eta_i^{Q}}{1-2\eta_i^{Q}}\right)\right] = \epsilon \tag{4}$$

To derive the optimal series, we model the problem using the continuous representations $\eta(x)$ and $\mathcal{L}(\eta(x))$. This simplifies the analysis using the integral calculus.

Definition 4. *Continuous functions $\eta:\mathbb{R}^+\to(0,0.5)$ and $\mathcal{L}:\eta(x)\to\frac{1}{2}\log(1-2\eta(x))\,(1+2\eta(x))$ are defined to be, the continuous representation of η_i, and its corresponding continuous loss function obtained by interpolating a continuous function over the sequences of $<\eta_i^{\mathcal{A}},\eta_i^{Q}>$.*

Moreover, we make two reasonable assumptions:

1) The interpolation error to be negligible.
2) The function $\eta(x)$ is twice continuously differentiable, indicating that it belongs to the class C^2.

Total Privacy over Infinite Iterations. Under this continuous model, we derive key relationships between the total privacy budget ϵ and the overall utility loss L in Remark 5.1 and Theorem 5.3.

Remark 5.1 (Proof in Appendix B.2)**.** *For the overall probability loss $L = \int_1^{\infty}\mathcal{L}(x)\,dx$, there exist constants ζ, m, M such that*

$$L \geq \frac{1}{4\zeta}\left(Li_2(M^2) - Li_2(m^2) + 4Li_2(m) - 4Li_2(M)\right) \tag{5}$$

where $Li_2(\cdot)$ is the poly-logarithm Spence's function.

Theorem 5.3 (Proof in Appendix B.3)**.** *The total privacy bound of DPI over infinite time slots is*

$$\epsilon \geq \frac{2\Delta_Q}{\mu\zeta}\left(Li_2(M^2) - Li_2(m^2)\right) \tag{6}$$

Since the $Li_2(M^2)$ is the poly-logarithms of the Spence's function, the overall privacy loss is growing logarithmically. Thus, Theorem 5.3 signifies that the privacy loss is converging and effectively bounded in DPI.

Utility Loss. We now present our final result: a series that maximizes the accuracy of PDF estimation in DPI.

Theorem 5.4 (Proof in Appendix B.4)**.** *Given total privacy budget ϵ, DPI mechanism achieves an optimal utility (in a probability loss sense) with the series $\eta_t = \frac{1}{2}\cdot\frac{e^X-1}{e^X+1}$:*

$$|L| \geq \frac{1}{4\zeta}\left|\frac{\epsilon}{C} - \frac{2\pi^2}{3} + 4Li_2\left(\sqrt{Li_2^{-1}\left(\frac{\pi^2}{6}-\frac{\epsilon}{C}\right)}\right)\right|. \tag{7}$$

where $Li_2^{-1}(\cdot)$ represents the inverse of poly-logarithm of the Spence's function, $X = \frac{1-(Li_2^{-1}(\frac{\pi^2}{6}-\frac{\epsilon}{C}))^t}{t^2|\zeta|}$ $|\zeta| = \sup_{\{t=2,3,\dots\}}\{e^{|\eta_{k-1}-\eta_k|}\}$ and $C = \frac{\Delta_Q}{\mu|\zeta|}$.

DPI leverages the inverse of the poly-logarithm of the Spence's function which can be represented as a converging infinite series to guarantee the loss bound. As t grows, the utility loss will converge. The converging series provides a tight utility guarantee in Theorem 5.4.

Summary. Our theoretical analyses have shown that DPI can provide strong cumulative privacy and high utility for infinite data streams. Specifically, we prove DPI's overall privacy budget grows logarithmically with iterations (Theorem 5.3). This allows continuous operation under a fixed total privacy budget ϵ. We also derive an upper bound on DPI's total error over time (Theorem 5.2), quantifying its utility. Furthermore, we obtain the optimal strategy for η_t across iterations to maximize utility under ϵ (Theorem 5.4). These results show that DPI provably converges to O(ϵ) error for answering pre-defined queries under privacy budget ϵ.

6. Experimental Evaluations

6.1. Experimental Setting

Experimental Datasets. We conduct all the experiments on three real-world and some synthetic streaming datasets to evaluate the performance of our DPI. Table 2 shows the

characteristics of the datasets (the time period is partitioned into specific numbers of equal-length time slots).

TABLE 2. CHARACTERISTICS OF DATASETS (AFTER PRE-PROCESSING)

Dataset	User #	Domain Size	Slot #	Period
COVID-19	$48,925$	110	$1,000$	6 month
Network Traffic	$5,074,413$	$65,534$	$2,700$	1 hour
USDA Production	$1,695,038$	$111,989$	$2,700$	1960-2022

COVID-19 Dataset [30] provides a chronological sequence of COVID-19 confirmed cases, deaths, and recoveries, allowing researchers to analyze the trend and pattern of the pandemic over time. Moreover, the attributes are disaggregated by 110 countries and subregions aiding in regional and comparative studies, enhancing the understanding of the geographical variation and spread of the virus.

Network Traffic Dataset [31] is used to study a variety of real-world DDoS attacks. This dataset contains network traffic such as source and destination IP addresses, etc. $65,534$ source IP addresses and their network traffic records were extracted for our experiments.

USDA Production Supply and Distribution Dataset [32] collects the agricultural production data for 300 commodities with over 100 attributes, including $111,989$ categories of $1,695,038$ production records from 1960 to 2022.

Applications for Evaluation. We next introduce three real-world applications for evaluating the performance of DPI.

Statistical Queries. Since our framework DPI could release the data distribution in each time slot, we first evaluate the utility for querying the item distribution: the difference between the original data's item distribution and the DPI output using the MSE and KL divergence metrics.

Anomaly Detection. In addition to statistical queries, the data distribution released by DPI facilitates various downstream analyses [33], [34], [35], [36], [37], [38], [39], [40]. In our experiments, we conduct experiments on anomaly detection (*most existing methods cannot support such analysis*) by adopting a distribution-based anomaly detection method [41] to detect the anomalies with the DPI output and evaluate the accuracy.

Recommender System. DPI also enhances recommender systems by providing privacy guarantees without compromising accuracy. Singular Value Decomposition (SVD) algorithms [42], [43] used in recommender systems can investigate the relationships between items by using the distribution of users and items. In our experiments, DPI enables precise suggestions by providing differentially private data distributions to the SVD algorithm.

Benchmarks. We have shown that existing methods violate the privacy bound ϵ after a few time slots (see Appendix E.1). They are incomparable with DPI in our infinite settings due to their limitations (e.g., unbounded privacy, sensitivity assumption, event-level differential privacy).

6.2. Performance on Hyperparameters

The utility of DPI is influenced by parameters such as ϵ, λ (error bound for synopsis), μ (error bound for avoiding overfitting), and ζ (decay rate of the decaying series), which are all related to the reweighting phase. To explore how they affect the utility in real-time disclosure, we choose the COVID-19 dataset to observe the utility trends with varying hyperparameters. The privacy requirement ϵ ranges from 0.1 to 10, and ζ varies from 0.1 to 0.5. λ is set in the range 0.05 to 0.5 and μ varies from 0.05 to 0.3.

We show the MSE and KL divergence regarding ϵ for $1,000$ time slots in Figure 3. Figures 3(a), 3(c), 3(e), 3(g), 3(i), and 3(k) are the MSE and KL divergence with different λ, μ, and ζ values for accumulative data distribution disclosure (from time slot 1 to $1,000$). Figures 3(b), 3(d), 3(f), 3(h), 3(j), and 3(l) are the MSE and KL divergence with different λ, μ, and ζ values for instantaneous data distribution disclosure (time slot $1,000$).

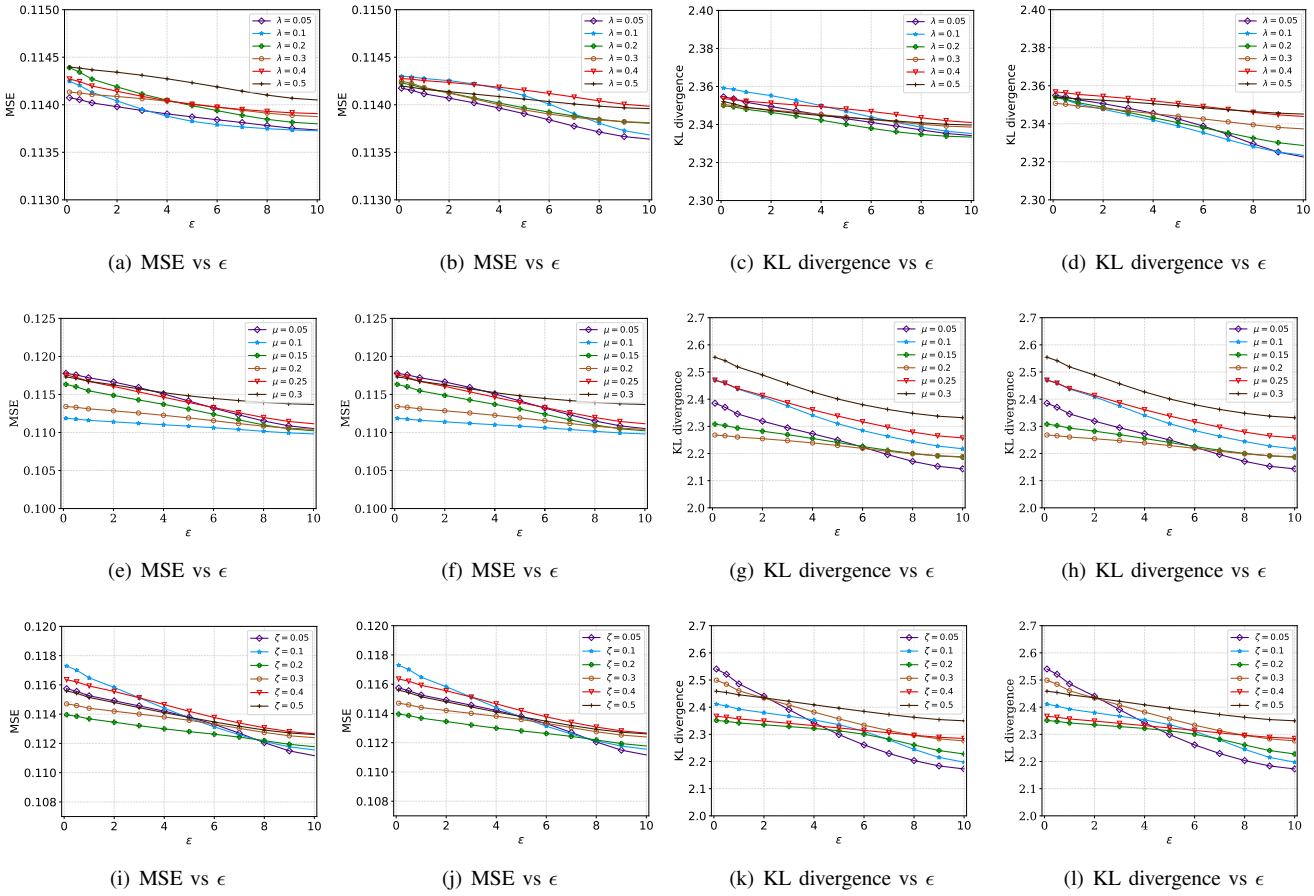
First, Figure 3 demonstrates the MSE and KL divergence by varying ϵ for accumulative data distribution disclosure and instantaneous data distribution disclosure. With an increasing ϵ, the MSE and KL divergence both decrease, demonstrating that the DPI output is closer to the true distribution (privacy/utility tradeoff). Second, Figures 3(c) and 3(d) show the KL divergence decreases faster when λ is very small (purple line) since the parameter λ evaluates the error of query based on the sampled synopsis and a small λ means that the synopsis is more accurate. Thus, the output results of DPI tend to be accurate. Furthermore, the smaller additional error bound μ (for avoiding overfitting) indicating the confidence for the universal synopsis output also affects the performance of DPI. Figures 3(g) and 3(h) demonstrate that we can have good utility with small KL divergence given small μ (e.g., 0.05 or 0.1). In our settings, μ is set to be less than half of λ. Finally, Figures 3(k) and 3(l) validate that small ζ (e.g., 0.05, 0.1, or 0.2) returns higher accuracy (since smaller ζ produces larger η, leading to more accurate synopses).

Similarly, we can draw similar observations from Figures 3(a) and 3(b), 3(e) and 3(f), 3(i) and 3(j) (with the MSE metric). This confirms that the effect of different parameters on DPI aligns with our theoretical studies.

6.3. Utility Evaluation

Statistical Queries. We consider two queries: sum and mean. We evaluate the KL divergence of total $1,000$ time slots with a varying ϵ and present the results in Figures 4(a) and 4(b). The results show that the utility of DPI improves as ϵ increases. Even with a small ϵ, DPI ensures low MSE values, e.g., ~ 0.12 when $\epsilon = 0.1$. In addition, we also evaluate the MSE and KL divergence in each of the $1,000$ time slots, as shown in Figures 4(c) and 4(d). The MSE values for all statistical queries are quite low, up to 0.16.

Anomaly Detection. Another application focuses on anomaly detection in the Network Traffic dataset. In this task, we will apply the Histogram-based Outlier Score (HBOS) [41] (equal to a distribution-based anomaly detection technique) and isolation forest [44] to identify anomalies. The HBOS value reflects the rarity of data in the learned



(a) MSE vs ϵ (b) MSE vs ϵ (c) KL divergence vs ϵ (d) KL divergence vs ϵ

(e) MSE vs ϵ (f) MSE vs ϵ (g) KL divergence vs ϵ (h) KL divergence vs ϵ

(i) MSE vs ϵ (j) MSE vs ϵ (k) KL divergence vs ϵ (l) KL divergence vs ϵ

Figure 3. Average MSE and KL divergence regarding ϵ for $2,700$ time slots. Figures (a), (c), (e), (g), (i), and (k) are MSE and KL divergence with different λ, μ, and ζ values for accumulative data distribution disclosure. Figures (b), (d), (f), (h), (j), and (l) are the MSE and KL divergence with different λ, μ, and ζ values for instantaneous data distribution disclosure.

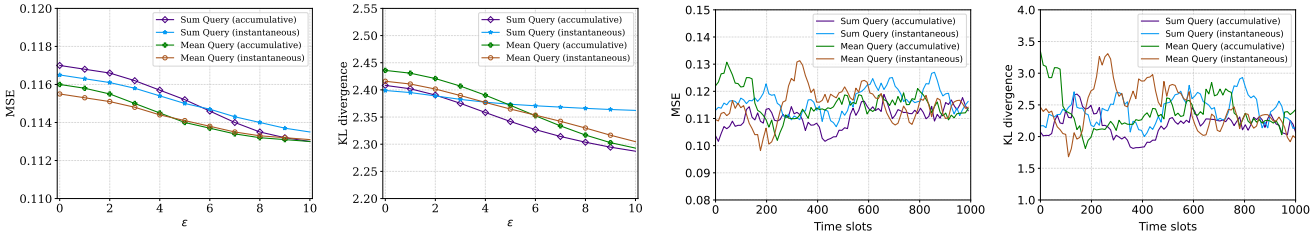
distribution and the extent of deviation of certain features from this distribution. The network monitor can quickly identify and report any anomalies from incoming packets, along with their severity, to the respective tenants.

We conduct experiments on the Network Traffic dataset and define the items with a score over a threshold as anomalies. DPI is expected to detect the anomalies from its real-time disclosures. We adopt Precision and Recall as the evaluation metrics compared to the non-private results, as shown in Figure 5. The Precision and Recall of both HBOS and Isolation Forest increase linearly with a growing ϵ. Moreover, anomaly detection can be accurately performed by dynamically tracking the new data over a large number of time slots (as observed from the fluctuated Precision and Recall scores). Finally, anomaly detection over all the time slots (accumulative) tends to show relatively better results than that in specific time slots (instantaneous).

Recomender System. For recommender systems, we utilize the SVD algorithm [42], [43], which capitalizes on data distribution for delivering recommendations by efficiently managing dimensionality reduction and highlighting essential features. It decomposes a user-item matrix into three constituent matrices, capturing the interaction between users and items. By reducing the dimensionality, SVD effectively uncovers latent features indicative of user preferences. Then, SVD leverages the data distribution to recommend items that align with users' interests based on their past behavior. We also test the utility with the K-means [45], [46] algorithm. Figure 6 shows that both Precision and Recall increase as ϵ increases. They can be very high by considering the non-private results as the ground truth (e.g., 90%+).

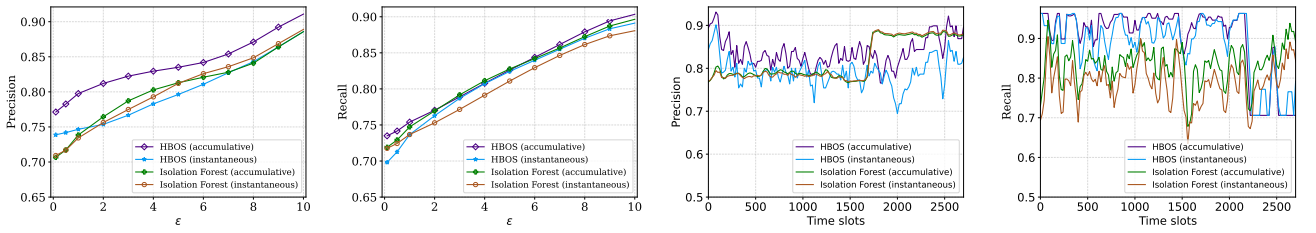
6.4. Highly Dynamic Data Distributions

To evaluate DPI on highly dynamic data streams, we generated a synthetic dataset over $3,000$ time slots. For each time slot, we generate a synthetic dataset by initializing 100 items in each dataset and sampling their counts with the Gaussian distributions (mean 100 and variance randomly chosen from $1, 4, 9, 16, 25, \ldots, 100$). Figure 9 shows the true values in the synthetic datasets and the DPI outputs under different ϵ (since there are numerous values in each time slot, we plot the median values in the figure). Then, we can observe that the DPI outputs are close to the true values by well preserving the original data distribution.



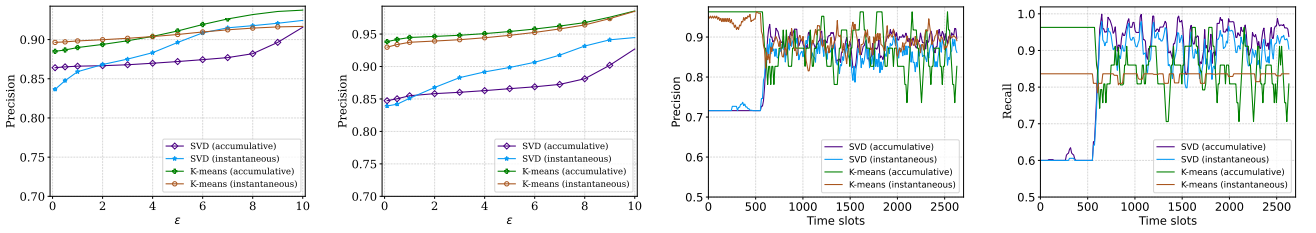
(a) MSE vs ϵ (b) KL divergence vs ϵ (c) MSE (moving average) vs t (d) KL div. (moving average) vs t

Figure 4. Average MSE and KL divergence of statistical queries for ϵ and t across $1,000$ time slots on COVID-19 dataset. (a), (b): Average MSE and KL divergence with varying ϵ. (c), (d): Average MSE and KL divergence over t when ϵ is 2.



(a) Precision vs ϵ (b) Recall vs ϵ (c) Precision (moving average) vs t (d) Recall (moving average) vs t

Figure 5. Average Precision and Recall for anomaly detection over $2,700$ time slots on Network Traffic dataset. (a), (b): anomaly detection with different ϵ. (c), (d): anomaly detection in time slot t ($\epsilon = 2$). Due to unsupervised learning on unlabeled data, non-private results are treated as the ground truth.



(a) Precision vs ϵ (b) Recall vs ϵ (c) Precision (moving average) vs t (d) Recall (moving average) vs t

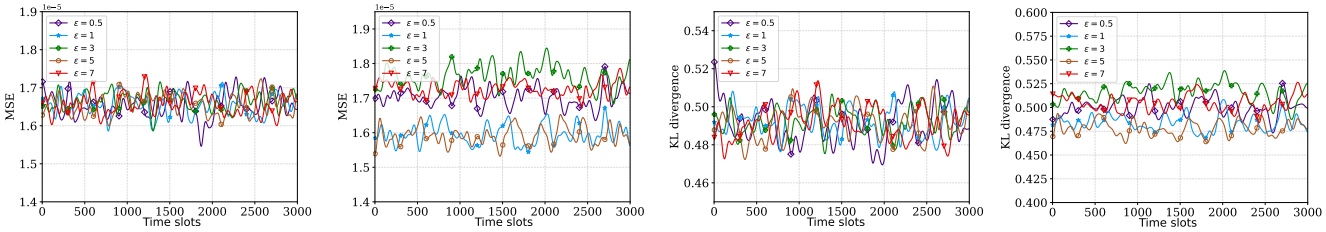
Figure 6. Average Precision and Recall for recommender system over $2,700$ time slots on USDA dataset. (a), (b): Precision and Recall vs ϵ. (c), (d): Precision and Recall vs time slot t ($\epsilon = 2$). Due to unsupervised learning on unlabeled data, non-private results are treated as the ground truth.

Furthermore, Figure 7 shows the MSE and KL divergence of DPI outputs over time under different ϵ. From the plots, we observe the values of MSE and KL divergence are very small even in case of small ϵ (e.g., 0.5 and 1). Moreover, although the distribution changes significantly over time, the MSE and KL divergence of DPI are very stable. The boxplots in Figure 8 show the lower quantile (LQ), upper quantile (UQ), and median (M) of the DPI output densities. The output density here refers to the distribution of DPI outputs in terms of their frequency and spread. The spread of the distributions remains stable over time.

Finally, we conduct experiments on 20 more synthetic datasets and different $\epsilon \in [0.5, 10]$ (see Figure 15 in Appendix E.2). First, we generate 10 distinct synthetic datasets by initializing $1,000$ items in each dataset and sample their counts with the Gaussian distributions (mean $1,000$ and variances $1, 4, 9, 16, 25, \ldots, 100$ for 10 datasets, respectively). Second, we generate another 10 distinct synthetic datasets with the domain size $10,000$ and similar settings. Figure 15 further proves the stability and effectiveness of DPI across varying data distributions and domain sizes.

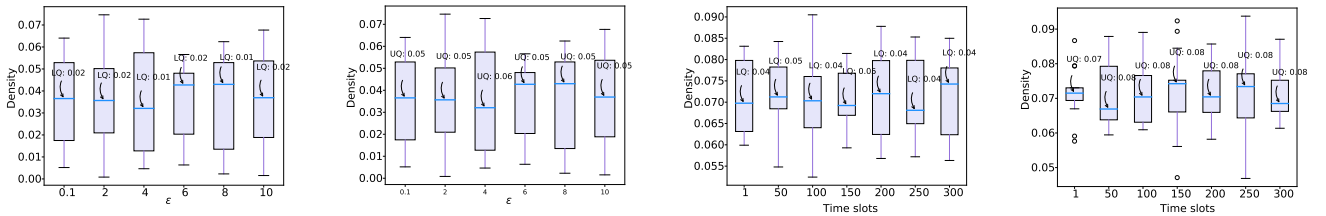
7. Discussion

Boosting and DPI. Boosting is a powerful technique in machine learning that combines multiple weak models to produce a stronger and more robust model. In the case of DPI, boosting is utilized to enhance privacy and accuracy through the continuously updated streaming data in each dynamic batch. In particular, we only run T boosting rounds in the first batch to get a good sampler PDF. Subsequently, DPI will continuously update the synopses sampling distribution in each time slot to the underlying data distribution with privacy preservation. Upon that, the efficiency of DPI can be significantly improved. DPI can also be modified to utilize boosting technique for each round. In this way, the accuracy can be better but cost more privacy budget.



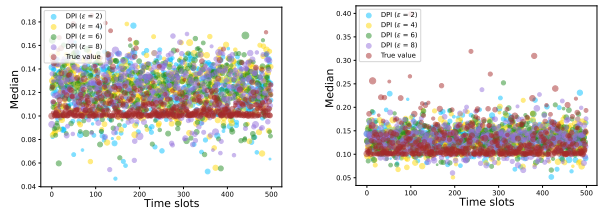
(a) MSE vs t (b) MSE vs t (c) KL div. vs t (d) KL div. vs t

Figure 7. Average MSE and KL divergence over $1,000$ time slots. (a) (b): MSE vs t (varying ϵ). (c) (d): KL divergence vs t (varying ϵ).



(a) Density vs ϵ (b) Density vs ϵ (c) Density vs t (d) Density vs t

Figure 8. Distribution spread of DPI output on the synthetic dataset with changing distribution. (a) (b): DPI output vs ϵ. (c) (d): DPI output vs t. The boxplots show the lower quantile (LQ), upper quantile (UQ), and median (M) of the DPI output densities. The spread of the distributions remains stable.



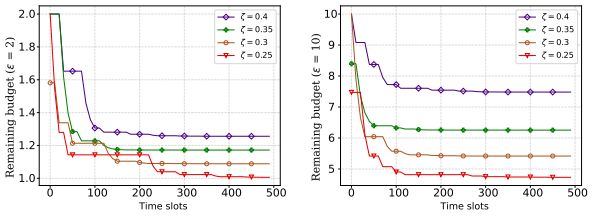
(a) Median values vs t (b) Median values vs t

Figure 9. Real-time DPI outputs and true data (median).

Complexity Analysis. The complexity analysis for DPI focuses on the primary computational components: the Boosting Algorithm, and the 0-DP Synopsis Generation algorithm. In DPI, the Boosting Algorithm updates and normalizes distribution synopses with a complexity of $O(n)$, and the 0-DP Synopsis Generation algorithm generates synopses through probabilistic sampling also with a complexity of $O(n)$. The overall time complexity $O(n)$ also aligns well with the standard streaming algorithms (over time series). Moreover, DPI consumes memory proportional to the domain size (which is tolerable even for large domains).

No Numerical Overflow for Extremely Tiny Budgets. The randomization in DPI is based on "DP-Boosting", which effectively translates extremely tiny budgets into negligible or even no reweighting. Thus, different from noise-additive DP mechanisms (e.g., Laplace and Gaussian), extremely tiny budgets in DPI do not entail any numerical overflow.

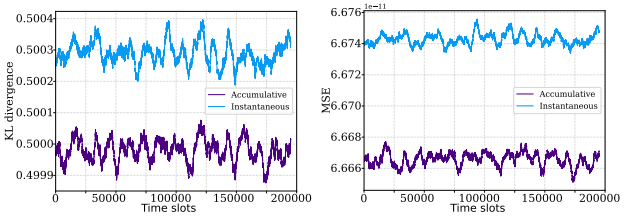
Limitations. First, DPI presents a new privacy-utility trade-off compared to traditional noise-additive mechanisms. Utility loss manifests as ineffective tracking of new stream data, potentially in extreme cases of dynamic data over extremely long periods, where privacy budget depletes, possibly necessitating DPI system restart for optimal utility with a renewed budget. However, DPI may not necessarily need to restart in practice for two reasons: (1) our random budget allocation ensures that budgets are not depleted even after a large number of time slots (Figure 10 demonstrates that budgets are adequate after 500 time slots), and (2) given extremely tiny privacy budgets, DPI maintains low MSE and KL divergence for different data distributions over extremely long periods. Figure 11 shows that the results are still stable even after $200,000$ time slots (no need to restart), evaluated on the synthetic data generated in Section 6.4. Indeed, we cannot find extreme cases of the low utility of DPI in our extensive empirical studies.



(a) Remaining budget vs t (b) Remaining budget vs t

Figure 10. Remaining budget for DPI with the random privacy budget allocation: (a) $\epsilon = 2$, and (b) $\epsilon = 10$.

Second, DPI outputs data as a series of PDFs instead of count histograms, suitable for most downstream analyses. Nevertheless, counts can be estimated from these PDFs by discretizing the domain into bins, calculating each bin's probability mass, and scaling these probabilities by the total data size. For instance, a 0.02 probability in a bin for



(a) Median (moving average) vs t (b) Median (moving average) vs t

Figure 11. Evaluation of DPI on 200, 000 time slots on synthetic datasets (a) KL divergence and (b) MSE: domain size 10, 000.

ages 15-20 in a million-record dataset implies an estimated 20,000 individuals in that age group. To privately disclose counts in certain applications, DPI only needs to assume the release of the non-private total data size at time slot t.

8. Related Work

Data Streaming with DP. Since the early studies on the private data streaming [47], [48], DP models were proposed to protect the streaming data. Ebadi et al. [49] proposed a personalized DP for dynamically adding records to the database. Meanwhile, Liu et al. [50] also proposed personalized DP with the weighted posterior sampling to reduce the extra Gaussian noise to the parameter space. To further apply DP to streaming data, many works [5], [8], [51], [52], [6], [7], [12] focus on continuously publishing statistics computed over events. Chan et al. [6] propose several methods that can handle binary streams of different users over potential infinite streams. Several works [7], [9], [10] follow a similar idea of applying partition algorithms to handle numeric values in the streaming data. However, event-level privacy is not strong enough to prevent sensitive data. Later, Kellaris et al. [11] proposed the w-event privacy to protect event sequence occurrence in w timestamps. This notion converges to user-level privacy when w is set to infinite, but the noise would be unbounded. Aiming to achieve user-level DP, FAST [13] was proposed to release real-time statistics without full DP infinite data stream.

Query Boosting with DP. Boosting has been widely used for improving the accuracy of learning algorithms [53]. There are also many algorithms that boost the DP results. In [54], they combined private learning with DP, using learning theory to comprehend database probabilities, potentially enhancing DP's distortion reduction. For general counting queries, Dwork et al. proposed a base synopsis generator [25], and also constructs an appropriate base synopsis generator for any set of low-sensitivity queries (not just counting queries) [22]. These well-designed synopsis generators can significantly boost the utility of DP results. Recall that our synopsis generator significantly differs from them to support infinite data streaming.

Data Streaming with LDP. LDP [20] involves users perturbing their data before sending it to an untrusted server for aggregation, posing challenges in privacy accumulation for streaming data with limited utility (e.g., a sequence of locations [55]). Joseph et al. [56] proposed a framework for continuous data sharing under LDP, consuming privacy budget based on distribution changes rather than collection periods to reduce the overall privacy expenditure. Wang et al. [10] proposed an algorithm using the Exponential mechanism with a quality function to publish a stream of real-valued data under both centralized and LDP. Li et al. [57] proposed an LDP approach to heavy hitter detection on data streams with bounded memory. These works can provide event-level privacy protection. Bao et al. [58] proposed a novel correlated Gaussian mechanism for (ϵ, δ)-LDP on streaming data aggregation. However, CGM needs to update the privacy budget periodically. Compared to these, Ren et al. [59] then proposed a population division framework that not only avoids the high sensitivity of LDP noise to the budget division but also requires less communication. However, it can only provide the w-event privacy.

Applications. Differential privacy (DP) has been the de facto rigorous privacy solution for learning algorithms [60], [61], [62], [63], [64], [65], [66], [67]. McSherry et al. [68] applied DP to provide personal recommendations. Chen et al. [69] proposed to publish the rating matrix of the source domain with DP guarantee. Feng et al. [70] proposed a topic privacy-relevance parameter method for top-k recommendations. Okada et al. [71] used three queries on statistical aggregation on outliers to detect the occurrence of anomalous situations with differential privacy guarantee.

9. Conclusion

This paper addresses the significant challenges in an open problem: differential privacy for infinite data streams, which is crucial for various real-time monitoring and analytics applications. DP has been widely used to protect streaming data, but it has key limitations in terms of unbounded privacy leakage and sensitivity for ensuring user-level protection. We propose a novel solution, DPI, to effectively bound privacy leakage and enhance accuracy in real-time analysis on infinite data streams. We have conducted extensive theoretical studies to prove the convergence of privacy bound and the bounded errors (utility loss) for DPI over infinite data streams. We have also conducted comprehensive experiments to validate DPI on various real streaming applications and datasets.

Acknowledgments

The authors sincerely thank the anonymous shepherd and all the reviewers for their constructive comments and suggestions. This work is supported in part by the National Science Foundation (NSF) under Grants No. CNS-2308730, CNS-2302689, CNS-2319277, and CMMI-2326341. It is also partially supported by the Cisco Research Award, the Synchrony Fellowship, the National Key Research and Development Program of China under Grant 2021YFB3100300, and the National Natural Science Foundation of China under Grants U20A20178 and 62072395.

References

[1] P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, and J. M. Lopez-Soler, "Analysis and modelling of youtube traffic," *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 4, pp. 360–377, 2012.

[2] M. Thottan, G. Liu, and C. Ji, "Anomaly detection approaches for communication networks," in *Algorithms for next generation networks*. Springer, 2010, pp. 239–261.

[3] P. Ranjitha, "Streaming analytics over real-time big data," *Global Journal of Computer Science and Technology*, 2015.

[4] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proceedings of the forty-second ACM Symposium on Theory of Computing*, 2010, pp. 715–724.

[5] J. Bolot, N. Fawaz, S. Muthukrishnan, A. Nikolov, and N. Taft, "Private decayed predicate sums on streams," in *Proceedings of the 16th International Conference on Database Theory*, 2013, pp. 284–295.

[6] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 3, nov 2011. [Online]. Available: https://doi.org/10.1145/2043621.2043626

[7] Y. Chen, A. Machanavajjhala, M. Hay, and G. Miklau, "Pegasus: Data-adaptive differentially private stream processing," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, p. 1375–1388.

[8] C. Dwork, "Differential privacy in new settings," in *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2010, pp. 174–183.

[9] V. Perrier, H. J. Asghar, and D. Kaafar, "Private continual release of real-valued data streams," *arXiv preprint arXiv:1811.03197*, 2018.

[10] T. Wang, J. Q. Chen, Z. Zhang, D. Su, Y. Cheng, Z. Li, N. Li, and S. Jha, "Continuous release of data streams under both centralized and local differential privacy," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2021, p. 1237–1253.

[11] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," *Proc. VLDB Endow.*, vol. 7, no. 12, p. 1155–1166, aug 2014. [Online]. Available: https://doi.org/10.14778/2732977.2732989

[12] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Rescuedp: Real-time spatio-temporal crowd-sourced data publishing with differential privacy," in *The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[13] L. Fan and L. Xiong, "An adaptive approach to real-time aggregate monitoring with differential privacy," *IEEE Transactions on knowledge and data engineering*, vol. 26, no. 9, pp. 2094–2106, 2013.

[14] L. Fan, L. Xiong, and V. Sunderam, "Differentially private multi-dimensional time series release for traffic monitoring," in *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 2013, pp. 33–48.

[15] H. Wang, S. Xie, and Y. Hong, "Videodp: A flexible platform for video analytics with differential privacy," *Proc. Priv. Enhancing Technol.*, vol. 2020.

[16] H. Wang, Y. Hong, Y. Kong, and J. Vaidya, "Publishing video data with indistinguishable objects," in *Proceedings of the 23rd International Conference on Extending Database Technology*, 2020, pp. 323–334.

[17] B. Liu, S. Xie, H. Wang, Y. Hong, X. Ban, and M. Mohammady, "VTDP: privately sanitizing fine-grained vehicle trajectory data with boosted utility," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 6, pp. 2643–2657, 2021. [Online]. Available: https://doi.org/10.1109/TDSC.2019.2960336

[18] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, vol. 4978, 2008, pp. 1–19.

[19] Y. Hong, J. Vaidya, H. Lu, P. Karras, and S. Goel, "Collaborative search log sanitization: Toward differential privacy and boosted utility," *IEEE Trans. Dependable Secur. Comput.*, vol. 12, no. 5, pp. 504–518, 2015.

[20] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 1054–1067.

[21] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[22] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE, 2010, pp. 51–60.

[23] M. Hardt and G. N. Rothblum, "A multiplicative weights mechanism for privacy-preserving data analysis," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, 2010, pp. 61–70.

[24] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[25] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. Vadhan, "On the complexity of differentially private data release: efficient algorithms and hardness results," in *Proceedings of the 41st annual ACM Symposium on Theory of Computing*, 2009, pp. 381–390.

[26] Y. Wang, X. Li, X. Li, and Y. Wang, "A survey of queries over uncertain data," *Knowledge and information systems*, vol. 37, no. 3, pp. 485–530, 2013.

[27] M. A. Soliman, I. F. Ilyas, and K. C.-C. Chang, "Probabilistic top-k and ranking-aggregate queries," *ACM Transactions on Database Systems (TODS)*, vol. 33, no. 3, pp. 1–54, 2008.

[28] I. F. Ilyas, W. G. Aref, and A. K. Elmagarmid, "Supporting top-k join queries in relational databases," *The VLDB journal*, vol. 13, pp. 207–221, 2004.

[29] X. Li, Y. J. Kim, R. Govindan, and W. Hong, "Multi-dimensional range queries in sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 63–75.

[30] D. GAURAV, "Real-time covid 19 data," 2023. [Online]. Available: https://www.kaggle.com/datasets/gauravduttakiit/covid-19

[31] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*, 2019, pp. 1–8.

[32] Foreign Agricultural Service, Department of Agriculture, "Production, supply, and distribution database," http://apps.fas.usda.gov/psdonline, 2020. [Online]. Available: https://www.kaggle.com/datasets/jeffersongranado/usdapsd?rvi=1

[33] A. Coluccia, A. D'Alconzo, and F. Ricciato, "Distribution-based anomaly detection via generalized likelihood ratio test: A general maximum entropy approach," *Computer Networks*, vol. 57, no. 17, pp. 3446–3462, 2013.

[34] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[35] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International conference on learning representations*, 2018.

[36] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "Padim: a patch distribution modeling framework for anomaly detection and localization," in *ICPR International Workshops and Challenges*. Springer, 2021, pp. 475–489.

[37] H. S. Asif, P. A. Papakonstantinou, and J. Vaidya, "How to accurately and privately identify anomalies," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2019, pp. 719–736.

[38] J. A. Giraldo, A. A. Cárdenas, M. Kantarcioglu, and J. Katz, "Adversarial classification under differential privacy," in *NDSS*, 2020.

[39] M. Mohammady, H. Wang, L. Wang, M. Zhang, Y. Jarraya, S. Majumdar, M. Pourzandi, M. Debbabi, and Y. Hong, "Dpoad: Differentially private outsourcing of anomaly detection through iterative sensitivity learning," *arXiv preprint arXiv:2206.13046*, 2022.

[40] M. Du, R. Jia, and D. X. Song, "Robust anomaly detection and backdoor attack detection via differential privacy," *ArXiv*, vol. abs/1911.07116, 2020.

[41] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," *KI-2012: poster and demo track*, vol. 1, pp. 59–63, 2012.

[42] B. Sarwar, G. Karypis, J. Konstan, and J. T. Riedl, "Application of dimensionality reduction in recommender system-a case study," 2000.

[43] X. Zhou, J. He, G. Huang, and Y. Zhang, "Svd-based incremental approaches for recommender systems," *Journal of Computer and System Sciences*, vol. 81, no. 4, pp. 717–733, 2015.

[44] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, p. 3, 2012.

[45] Y. Zhang, N. Liu, and S. Wang, "A differential privacy protecting k-means clustering algorithm based on contour coefficients," *PloS one*, vol. 13, no. 11, p. e0206832, 2018.

[46] C. Yin, L. Shi, R. Sun, and J. Wang, "Improved collaborative filtering recommendation algorithm based on differential privacy protection," *The Journal of Supercomputing*, vol. 76, pp. 5161–5174, 2020.

[47] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD Workshop*, 2003.

[48] S. Chandrasekaran and M. J. Franklin, "Streaming queries over streaming data," in *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*. Elsevier, 2002, pp. 203–214.

[49] H. Ebadi, D. Sands, and G. Schneider, "Differential privacy: Now it's getting personal," ser. POPL '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 69–81. [Online]. Available: https://doi.org/10.1145/2676726.2677005

[50] Z. Liu, Y.-X. Wang, and A. Smola, "Fast differentially private matrix factorization," in *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, p. 171–178.

[51] D. Mir, S. Muthukrishnan, A. Nikolov, and R. N. Wright, "Pan-private algorithms via statistics on sketches," in *Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2011, pp. 37–48.

[52] T.-H. H. Chan, M. Li, E. Shi, and W. Xu, "Differentially private continual monitoring of heavy hitters from distributed streams," in *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2012, pp. 140–159.

[53] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," in *Proceedings of the 11th annual conference on Computational learning theory*, 1998, pp. 80–91.

[54] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: The sulq framework," 06 2005, pp. 128–138.

[55] H. Wang, H. Hong, L. Xiong, Z. Qin, and Y. Hong, "L-SRR: local differential privacy for location-based services with staircase randomized response," in *Proceedings of the ACM Conference on Computer and Communications Security*, 2022, pp. 2809–2823.

[56] M. Joseph, A. Roth, J. Ullman, and B. Waggoner, "Local differential privacy for evolving data," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[57] X. Li, W. Liu, J. Lou, Y. Hong, L. Zhang, Z. Qin, and K. Ren, "Local differentially private heavy hitter detection in data streams with bounded memory," in *Proceedings of the 2024 ACM SIGMOD International Conference on Management of Data*, 2024.

[58] E. Bao, Y. Yang, X. Xiao, and B. Ding, "Cgm: an enhanced mechanism for streaming data collection with local differential privacy," *Proceedings of the VLDB Endowment*, vol. 14, no. 11, pp. 2258–2270, 2021.

[59] X. Ren, L. Shi, W. Yu, S. Yang, C. Zhao, and Z. Xu, "Ldp-ids: Local differential privacy for infinite data streams," in *Proceedings of the 2022 International Conference on Management of Data*, 2022, pp. 1064–1077.

[60] N. Papernot, "Machine learning at scale with differential privacy in TensorFlow," in *2019 USENIX Conference on Privacy Engineering Practice and Respect*, 2019.

[61] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, p. 308–318.

[62] J. Vaidya, B. Shafiq, A. Basu, and Y. Hong, "Differentially private naive bayes classification," in *2013 IEEE/WIC/ACM International Conferences on Web Intelligence*, 2013, pp. 571–576.

[63] N. Wu, F. Farokhi, D. Smith, and M. A. Kaafar, "The value of collaboration in convex machine learning with differential privacy," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 304–317.

[64] M. Mohammady, S. Xie, Y. Hong, M. Zhang, L. Wang, M. Pourzandi, and M. Debbabi, "R2dp: A universal and automated approach to optimizing the randomization mechanisms of differential privacy for utility metrics with no known optimal distributions," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 677–696.

[65] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[66] A. Friedman and A. Schuster, "Data mining with differential privacy," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 493–502.

[67] H. Wang, J. Sharma, S. Feng, K. Shu, and Y. Hong, "A model-agnostic approach to differentially private topic mining," in *The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1835–1845.

[68] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the netflix prize contenders," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, p. 627–636.

[69] C. Chen, H. Wu, J. Su, L. Lyu, X. Zheng, and L. Wang, "Differential private knowledge transfer for privacy-preserving cross-domain recommendation," in *Proceedings of the ACM Web Conference*, 2022, p. 1455–1465.

[70] T. Feng, Y. Guo, and Y. Chen, "A differential private collaborative filtering framework based on privacy-relevance of topics," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, 2016, pp. 946–951.

[71] R. Okada, K. Fukuchi, and J. Sakuma, "Differentially private analysis of outliers," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2015.

[72] S. Bloch, *Higher regulators, algebraic K-theory, and zeta functions of elliptic curves*. American Mathematical Soc., 2000, no. 11.

[73] P. Mateev, "On the entropy of the multinomial distribution," *Theory of Probability & Its Applications*, vol. 23, no. 1, pp. 188–190, 1978. [Online]. Available: https://doi.org/10.1137/1123020

[74] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th computer security foundations symposium (CSF)*. IEEE, 2017, pp. 263–275.

Appendix A.
DP with AdaBoosting [22]

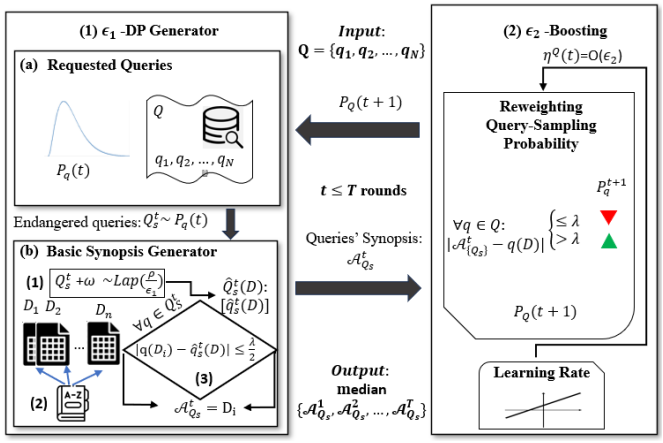


Figure 12. The architecture of DP AdaBoosting [22]

Appendix B.
Proofs

B.1. Proof of Theorem 5.2

As stated in Claim 4.3 in [22], the probability of inaccuracy produced by a DP boosting algorithm can be described in terms of (η, β)-base learners. In DPI, we can analogously consider $(\eta_i^{\mathcal{A}(q,t)}, \eta_i^{Q}, \beta)$-base learners at each iteration i.

After t rounds of DP boosting, the probability of inaccuracy in the DPI framework is upper-bounded by $\prod_{i=1}^{t} \sqrt{\left[(1-4(\eta_i^Q)^2)\,(1-4(\eta^{\mathcal{A}} i)^2)\right]}$. By taking the natural log of this bound, a version similar to the privacy bound (the sum of natural logs) is obtained. Specifically,

$$\log\left[\mathbb{P}(\text{bad outcome})\right] \leq \frac{1}{2}\sum i = 1^{N\to\infty} \log\left(1-2\eta_i\right)\left(1+2\eta_i\right)$$

Thus, this completes the proof. □

B.2. Proof of Remark 5.1

Our construction involves a decreasing sequence of privacy budgets, thus proving that η_i must also decrease. We assume $\eta(x) < 0.5$ is a converging function of at least an exponential order ($exp(^{-\zeta \cdot x})$). Then, we have

$$\eta'(x) \geq -\zeta\eta(x), \quad \forall x \in \mathbb{R}^+ \tag{8}$$

where $\zeta = |\inf_{x\in[1,\infty)} \frac{d\eta(x)}{dx}|$. By replacing $2\eta(x)$ with u, and for constants $m = \min_{x\geq 1}(\eta(x))$ and $M = \max_{x\geq 1}(\eta(x))$, we obtain

$$|L| \geq -\frac{1}{4}\int_m^M \frac{\log[(1+u)(1-u)]}{\zeta u} du$$

but given that the Spence's function [72] is defined as $Li_2(z) = -\int_0^z \frac{\log[(1-u)]}{u} du$, for $|z| \leq 1$, we have

$$|L| \geq \frac{1}{4\zeta}\left(Li_2(M^2) - Li_2(m^2) + 4Li_2(m) - 4Li_2(M)\right)$$

Here, we have utilized the following two results:

$$\int_0^z \frac{\log[(1+u)]}{u} du = -Li_2(-z) \tag{9}$$
$$Li_2(z) + Li_2(-z) = \tfrac{1}{2}Li_2(z^2)$$

Thus, this completes the proof. □

B.3. Proof of Theorem 5.3

According to Equation 4, the continuous format of DPI utilizes the following privacy budget.

$$\epsilon = \frac{4}{\mu}\int_0^\infty \log\left[\frac{1+2\eta(x)}{1-2\eta(x)}\right] dx,$$

By following a similar process as that described in the Remark 5.1, we will have

$$\epsilon \geq \frac{4}{\zeta\mu}\int_m^M \frac{\log\left[\frac{1+u}{1-u}\right]}{u} dx.$$

Finally, by utilizing the two results in Equation 9, we have

$$\epsilon \geq \frac{2}{\mu\zeta}\left(Li_2(M^2) - Li_2(m^2)\right).$$

Thus, this completes the proof. □

B.4. Proof of Theorem 5.4

By establishing lower bounds for both privacy and accuracy loss, we can determine the optimal series. Thus, our optimization problem can be succinctly stated as follows.

$$\min_{0\leq m<M\leq 1} \tfrac{1}{8\zeta}\left(Li_2(M^2) - Li_2(m^2) + 4Li_2(m) - 4Li_2(M)\right)$$
$$\text{w.r.t. } \tfrac{2}{\mu\zeta}\left(Li_2(M^2) - Li_2(m^2)\right) = \epsilon, \tag{10}$$

but Spence's function has the following infinite series form:

$$Li(z) = \sum_{t=1}^{\infty} \frac{z^t}{t^2}$$

Therefore, we have $Li_2(M) - Li_2(m) = \sum_{t=1}^{\infty} \frac{M^t - m^t}{t^2}$. Due to

$$Li_2(M^2) - Li(m^2) = \sum_{t=1}^{\infty} \frac{M^{2t} - m^{2t}}{t^2} = O(\epsilon), \tag{11}$$

However, we note that minimizing *probability loss* is equivalent to if we maximize $m^t + M^t$. This is true because our objective function has two opposite sign terms $A = [Li_2(M^2) - Li_2(m^2)] > 0$ and $B = 4Li_2(m) - 4Li_2(M) < 0$, where B appears in A, and minimization is translated into maximizing $(m^t + M^t)$ (using the sequence of *mean theroems*). Since Spence's function is strictly increasing, maximizing terms $m^t + M^t$, $\forall c$ means that $M = 1$.

Thus, when $M = 1 \Rightarrow C\left(Li(1) - Li\left(m^2\right)\right) = \epsilon$, we have

$$\Rightarrow Li\left(m^2\right) = Li(1) - \frac{\varepsilon}{C} = \frac{\pi^2}{6} - \frac{\epsilon}{C}, C = \frac{2}{\mu\zeta}$$
$$\Rightarrow m^2 = Li_2^{-1}(\frac{\pi^2}{6} - \frac{\epsilon}{C})$$

$$\Rightarrow m = \sqrt{Li_2^{-1}\left(\frac{\pi^2}{6} - \frac{\epsilon}{C}\right)} \tag{12}$$

Since each term in the series representation of the overall ϵ (Equation 11) has to be equal to the budget spent over the same index's disclosure, we have

$$\begin{aligned}
&\frac{2}{\mu\zeta} \cdot \sum_{t=1}^{\infty} \frac{M^{2t} - m^{2t}}{t^2} = \frac{4}{\mu} \log\left(\frac{1+2\eta_t}{1-2\eta_t}\right) \\
&\Rightarrow e^{\frac{1-m^{2t}}{2\zeta t^2}} = \frac{1+2\eta_t}{1-2\eta_t} \\
&\Rightarrow \eta_t = \frac{\left[e^{\frac{1-(Li_2^{-1}(\frac{\pi^2}{6} - \frac{\epsilon}{C}))^t}{t^2\zeta}} - 1\right]}{2 \cdot \left[e^{\frac{1-(Li_2^{-1}(\frac{\pi^2}{6} - \frac{\epsilon}{C}))^t}{t^2\zeta}} + 1\right]}
\end{aligned}$$

Thus, this completes the proof. □

Appendix C. Approximation of the Universe of PDFs

Algorithm 2: 0-DP Synopsis Generator

Input: streaming dataset D, size of dataset n, sample size N
Output: the universe of synopses $\mathcal{A}$

Initialize the quantization precision $p \in (0, 1)$ to make the corresponding precision level $n_p = \frac{1}{p} + 1$ equal to datset size n
$k \leftarrow$ distinct value number of data D
Initialize the synopses pool $\mathcal{A}$ as empty
for $i \in N$ **do**
    Set equal sampling probability for each distinct value in the domain $Prob = \frac{1}{k}$
    /* Sampling n times from k distinct values with $Prob = \frac{1}{k}$ and have a number of times each distinct value show among n times */
    $t \leftarrow$ Multinomial(n, k, $Prob$)
    /* Sum of P_i is 1 */
    $P_i \leftarrow t * p$
    $\mathcal{A}$.append(P_i)
return the universe of synopses $\mathcal{A}$

When n is sufficiently large, we can employ an asymptotic approximation to analyze the behavior. This approximation corresponds to the normal distribution limit of the Binomial distribution.

For large n, we have: $H \approx k - \frac{1}{2}\ln(2\pi ne) + \frac{1}{2}\sum_{i=1}^{k} \ln p_i$, where the natural logarithm is used for entropy calculation. Note that this approximation is not intended to provide an upper or lower bound, but rather an approximate form for large n. In the special case where the p_i are uniform, the expression simplifies to:

$$H \approx k - \frac{1}{2}\ln(2\pi ne) - \frac{k}{2}\ln k$$

While this is not an exact expression, it yields a highly accurate estimate of H when n is large.

Lemma C.1. *Let $H(\mathcal{A})$ denote the entropy of the constructed pool of synopses $\mathcal{A}$, and let H_{ideal} denote the entropy of the ideal multinomial PDF. As the number of samples drawn from the multinomial distribution, denoted as N, approaches infinity, the ratio $\frac{H(\mathcal{A})}{H_{ideal}}$ converges to 1.*

Proof. This proof is based on the concept that as the number of samples approaches infinity, the constructed pool of synopses becomes more representative of the ideal multinomial PDF. By constructing $\mathcal{A}$ from the multinomial distribution, which is designed to distribute the available probability units uniformly, the synopses generated cover a comprehensive range of possible outcomes. As a result, the entropy of $\mathcal{A}$ approaches the entropy of the ideal multinomial PDF. Therefore, the ratio $\frac{H(\mathcal{A})}{H_{\text{ideal}}}$ converges to 1. □

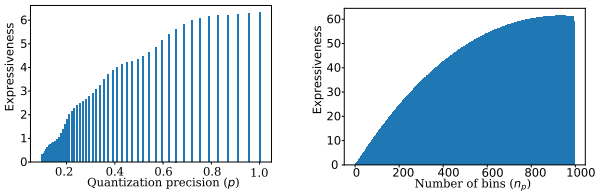
Theorem C.2. *Let $H(\mathcal{A}; n_p, k)$ denote the entropy of the constructed pool of synopses $\mathcal{A}$, given a precision level n_p (equivalent to $1/p$, where p is the quantization precision) and a domain size k. For sufficiently large values of N (number of samples drawn), $H(\mathcal{A}; n_p, k)$ approximates the entropy expression for the ideal multinomial PDF.*

Proof. The proof of Theorem C.2 involves comparing the entropy expression for the ideal multinomial PDF, denoted as H_{ideal}, with $H(\mathcal{A}; n, k)$. For a large value of n, the entropy of a heterogeneous multinomial PDF can be approximated as:

$$H_{\text{approx}}(N, k) \approx \frac{k-1}{2}\ln(2\pi Ne) - \frac{k}{2}\ln(k), \tag{13}$$

where N represents the number of samples drawn from the multinomial distribution [73].

By examining the entropy expression for the ideal multinomial PDF, which depends on k, and comparing it with $H(\mathcal{A}; n_p, k)$ for different values of n and k in the constructed pool of synopses, we can establish a relationship between the two. Specifically, for sufficiently large values of N, $H(\mathcal{A}; n_p, k)$ approaches $H_{\text{approx}}(N, k)$, indicating that the constructed pool of synopses accurately represents the ideal multinomial PDF. Figure 13 shows the information theory of 0-DP Synopsis Generation. □



(a) Expressiveness vs p (b) Expressiveness vs n_p

Figure 13. Information theory of 0-DP synopsis generation.

Appendix D. Random Budget Allocation (Alternative)

Random budget allocation with a non-depleting total budget output can also be achieved by dividing the total privacy budget ϵ into several ranges like *high*, *medium*, and *low*. In each time slot t, ϵ_t is randomly selected from one of these ranges. Refer to Algorithm 4 for details of this alternative approach. In essence, the random allocation of the privacy budget allows us to cater to diverse and changing data streams, ensuring the high utility of DPI while preserving privacy.

TABLE 3. EXPANDED SENSITIVITY AND PRIVACY LOSS (ACTUAL DP) FOR THE MEDIAN QUERY OF PACKETS IN EACH TIME SLOT (0.5 SECONDS) ON A NETWORK TRAFFIC DATASET. $\epsilon = 0.5$ IS USED TO GENERATE NOISE FOR EXISTING METHODS (ϵ IS ALWAYS BOUNDED BY 0.5 IN DPI).

	Sensitivity after 5 slots	Sensitivity after 50 slots	Actual DP after 5 slots	Actual DP after 50 slots
Chan et al. [6]	7	67	0.590	1.771
Chen et al. [7]	7	67	1.841	2.166
Perrier et al. [9]	7	67	1.734	2.248
Wang et al. [10]	7	67	1.674	1.972
DPI (ours)	2	2	0.001	0.005

Algorithm 3: Random Budget Allocation (RBA)

Data: available privacy budget space $\varepsilon_t = \{\epsilon_1, \epsilon_2, \ldots\}$
Data: tuning parameter $\Lambda = 10^8$
/* Λ is a large number tunable to scale of remaining epsilon values in ε_t */
/* sample from exponential distribution */
$S \sim \text{Exp}(\Lambda)$;
/* find the closest remaining privacy budget to S */
$\epsilon_t \leftarrow \arg\min_{\epsilon \in \varepsilon_t} |\epsilon - S|$;
/* update available privacy budget space */
$\varepsilon_{t+1} \leftarrow \varepsilon_t \setminus \{\epsilon_t\}$;
/* return the selected privacy budget */
return ϵ_t;

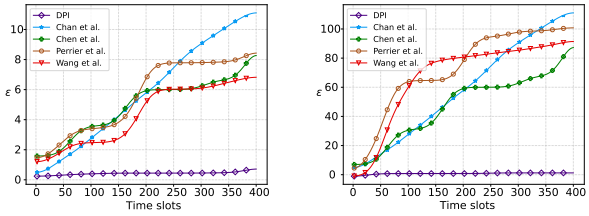
Algorithm 4: RBA: Alternative Approach

initialize queues $smallNumbers$, $mediumNumbers$, $largeNumbers$
define the ranges: $smallRange$, $mediumRange$, $largeRange$
for *each* i *in* $smallRange$, $mediumRange$, *and* $largeRange$ **do**
    correspondingQueue.enqueue(i)
while *true* **do**
    $randomQueue \leftarrow$ randomly choose from ($smallNumbers$, $mediumNumbers$, $largeNumbers$)
    if $randomQueue$ *is not empty* **then**
        $number \leftarrow randomQueue$.dequeue()
        return $number$

Appendix E. Additional Results

E.1. Privacy Loss of DPI vs Existing Methods

We additionally conduct experiments to compare the privacy loss of DPI with representative existing methods [7], [6], [10], [9]. Existing methods for differentially private data stream disclosures have several limitations. Most critically, they cannot preserve overall privacy loss in a bounded or converging manner. To prove this, we measure the expansion of sensitivity across the first 5 time slots and 50 time slots, respectively (in terms of the user-level DP protection) using the median query of packets in each time slot over a network traffic dataset in Table 3. In addition, we derive the privacy loss via the notion of Rényi differential privacy (e.g., deriving the Rényi differential privacy guarantees after injecting the noise and then converting the Rényi differential privacy to ϵ-DP [74]). Figure 14 further demonstrates that existing methods cannot bound the privacy over time. Thus, we do not benchmark DPI with them in the infinite stream settings.

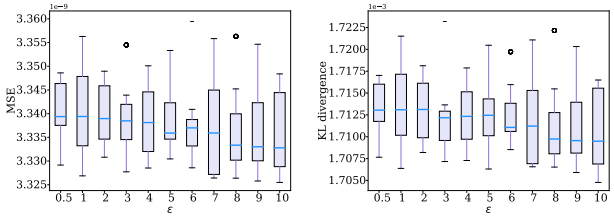


(a) 0.5 seconds each time slot (b) 5 seconds each time slot

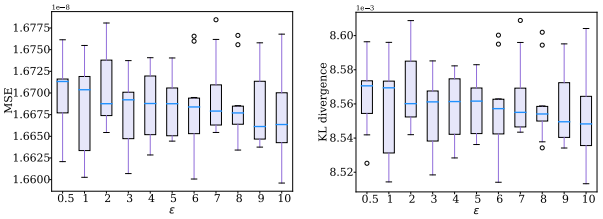
Figure 14. Total privacy loss of DPI and existing methods.

E.2. Additional Results on Synthetic Data

To further show that DPI can produce accurate output results on datasets with diverse characteristics, we conducted additional experiments on more synthetic data. Figure 15 shows the utility of DPI (w.l.o.g., using the accumulative query on the data distribution as an example) evaluated on additional 20 synthetic datasets and different privacy budgets ϵ varying from 0.5 to 10. Out of $594,000$ query results ($2,700$ time slots, 11 different ϵ, 10 datasets, 2 different domain sizes), DPI can still ensure low and stable MSE and KL divergence in all these different settings. These results demonstrate high utility on datasets with diverse characteristics.



(a) MSE vs t (b) KL divergence vs t



(c) MSE vs t (d) KL divergence vs t

Figure 15. Evaluation of DPI on 10 synthetic datasets (each box includes 10 results for a specific privacy bound ϵ: the average MSE/KL of $2,700$ time slots in each of the 10 datasets). (a) (b) MSE and KL divergence: domain size $1,000$. (c) (d) MSE and KL divergence: domain size $10,000$. Each experiment is repeated for 10 times, and the average results are plotted in the figures.

Appendix F.
Meta-Review

The following meta-review was prepared by the program committee for the 2024 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

F.1. Summary

DPI is a framework for handling infinite data streams with differential privacy and bounded privacy leakage.

F.2. Scientific Contributions

- Provides a Valuable Step Forward in an Established Field

F.3. Reasons for Acceptance

The paper provides a valuable step forward in an established field. Handling streaming data with differential privacy, particularly infinite streams, is a known open research problem. The paper proposes the combination of several novel techniques to bound privacy loss, including: data independent synopses, DP boosting, and budget allocation from a converging infinite series.