# Measurement of Local Differential Privacy Techniques for IoT-based Streaming Data

Sharmin Afrose
Department of Computer Science
Virginia Tech
Blacksburg, VA, USA 24060
Email: sharminafrose@vt.edu

Danfeng (Daphne) Yao
Deparment of Computer Science
Virginia Tech
Blacksburg, VA, USA 24060
Email: danfeng@vt.edu

Olivera Kotevska
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA 37830
Email: kotevskao@ornl.gov

***Abstract*— Various Internet of Things (IoT) devices generate complex, dynamically changed, and infinite data streams. An adversary can cause harm if they can access the user's sensitive raw streaming data. Therefore, protecting the privacy of the data streams is crucial. In this paper, we explore the de-facto privacy standard local differential privacy mechanisms for streaming data. We compare the techniques and report the advantages and limitations. We also present component (smoother, perturber) variations and show that combining distribution-based noise during perturbation provides more flexibility to the interested entity.**

## I. Introduction

The number of intelligent systems around us is growing rapidly. An example of these Internet of Things (IoT) based systems are smart home devices, health monitors, autonomous vehicles, and the smart grid. They are collecting data about our home activities, our health, where we travel, and our electricity usage, respectively. These technical means are constantly growing in power and sophistication. We will see even more rapid development with the widespread deployment of 5G wireless networks, which will provide high-speed data transfer and more precise location information. However, as these systems scale up, the need for privacy and security increases. Currently, we observe the deficiency to ensure meaningful data privacy guarantees to our citizens, institutions, and infrastructure. Thus, the scientific challenge of data privacy encompasses numerous issues including public safety, health, and national security.

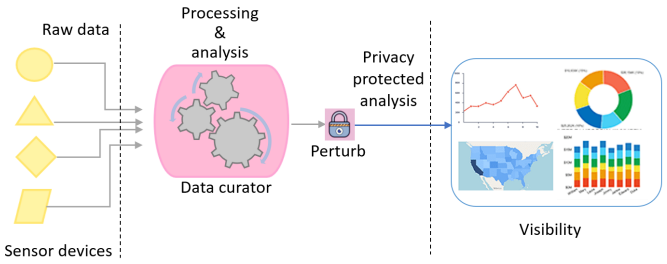
Privacy attacks take seemingly innocuous released information and use it to discern private details about individuals or national security [1]. Some attacks focus on identifying if an individual was part of the dataset [2] while others in identifying the sensitive information in the dataset [3] which are more common among biomedical-based systems. Other attacks are dedicated to reconstructing the model and interfering decision-making process [4]. However, there are many cases where the data was stolen before it reached the server or machine learning model. These cyber-security attacks are most common among music and video streaming applications such as Netflix, Hulu, Pandora, Spotify, but IoT-based solutions such as Fitbit, Apple Watch, Samsung SmartThings, are not an exception. Although in these cases security mechanisms lacked stronger guarantees, data privacy techniques can help in enforcing better protection.

Data privacy describes the practices which ensure that the data shared by customers is only used for its intended purpose and not used to cause harm. Many systems and platforms generate data that have sensitive properties. Sensitive properties are dependent on the type and purpose of the system such as critical infrastructure, smart home systems, or health monitoring devices. They collect data related to national security, personal security and human health. In the context of privacy in infrastructure it means users can find nearby IoT devices following privacy regulations or sharing the map of locations without presenting the actual coordinates but some estimate. In the context of system privacy can mean that when the state of the system is changed the actual value is not presented. The state of the system e.g. "on" or "off" can revel information if the system is working or not, in some scenarios such as light at home can even mean if someone is at home or not. This is considerate sensitive data because knowing the real value can be used by intruders to cause harm. Another example of sensitive data are the data collected by health monitoring devices such as blood pressure, sugar levels, heart rate which if someone get access to them can know what is the health status of the individual and use this data to sell products as an example. These examples show how sensitive data can be used to cause harm and having data privacy algorithms can prevent it.

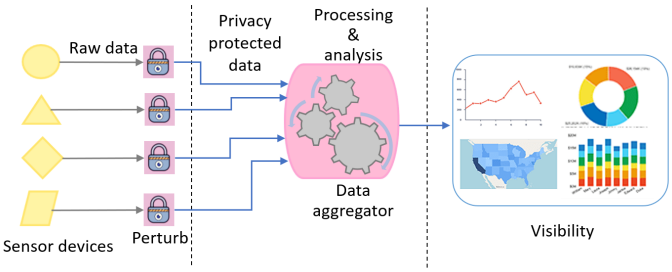
In this work, we looked into privacy techniques used for streaming applications such as IoT. We explore solutions based on local differential privacy (e.g., distribution-based techniques, randomized response-based, hash-based, etc.). We design the adaptation of several techniques (e.g., hash-based techniques) for streaming applications. Finally, we compare the methods and let the interested party know which methods are beneficial for them to use.

We also examine component variations for the distribution-based noise techniques. The reason for choosing the distribution-based noise technique is that it supports instantaneous reporting of noisy stream data. We show that a combination of different distribution-based noises widen the window for the privacy-utility trade-off.

The outline of the paper is organized as follows. First, we




(a) Differential privacy (DP)




(b) Local differential privacy (LDP)

Fig. 1: Workflow of stream data analysis under DP and LDP

describe the background knowledge in Section II. We explain the methodology in Section III. We present the evaluations and comparisons in Section IV. We report the related works in Section V. Finally we conclude in Section VI.

## II. Preliminaries

Differential privacy (DP), proposed by Dwork [5], provides strong mathematical privacy guarantees and it is defined as following:

**Definition II.1** ($\epsilon$-Differential privacy)**.** A randomized mechanism satisfies $\epsilon$-differential privacy where $\epsilon \geq 0$, if and only if datasets $D$ and $D'$, differing at most one value and $O \subseteq Range(F)$

$$Pr[F(D) \in O] \leq e^{\epsilon} Pr[F(D') \in O] \tag{1}$$

In DP, A data curator first collects the raw data and then performs an analysis. The analysis is then perturbed and released ((Fig. 1a)). The limitation of DP is that a trusted data curator is needed. Local differential privacy (LDP) is differential privacy in local settings. In LDP [6], the data is perturbed first before sending it to an aggregator for analysis (Fig. 1b). The advantage of LDP is that no trusted data aggregator is necessary.

**Definition II.2** ($\epsilon$-local differential privacy)**.** A randomized mechanism satisfies $\epsilon$-local differential privacy where $\epsilon \geq 0$, if and only if any pair of input values $v$ and $v' \in S$ and $O \subseteq Range(F)$

$$Pr[F(v) \in O] \leq e^{\epsilon} Pr[F(v') \in O] \tag{2}$$

In some cases we have joint distribution of deferentially private outputs that satisfies $\epsilon$-differential privacy, so sequential composition [7] is used.

**Theorem 1.** (Sequential composition). If a mechanism $F_i$ provides $\epsilon_i$-local differential privacy, a series of mechanisms on a data stream satisfies $\sum \epsilon_i$.

## III. Methodology

In this section, we describe selected local differential privacy algorithms for comparisons and component variations techniques.

### A. Local Differential Privacy (LDP) Techniques

*1) Distribution-based techniques:* For the distribution-based technique, we follow the DPLM privacy protection approach proposed by Hassan et. al. [8]. In their approach, they used the Laplace distribution-based noise mechanism. In our case, we apply four well-known distribution-based noise mechanisms (e.g., Laplace, Gaussian, Exponential, and Gamma). The raw power consumption data of every 10 minutes is perturbed by adding or removing random noise generated from different distributions. In addition, abnormal peaks are preserved to protect specific incidents (e.g., use of specific electronic instruments). The scale of the noise (e.g., sensitivity) is determined by the maximum allowed noise from the utility and user.


Fig. 2: Data stream perturbation (instant reporting)

The benefit of using distribution-based noise is that we interested third party can obtain instantaneous perturbed data as shown in Fig. 2. With this data, an interested party can calculate the frequency estimation (e.g., highest load in which hour), summation value (e.g., user's power consumption for the whole month), and others. The limitation is that we need to choose an optimal peak value (i.e., sensitivity) for the distribution-based noise.

*2) Randomized response-based techniques:* We consider three randomized response (RR) techniques for our experiment, i.e., Simplified RR, Memoized RR, Bloom RR. In the Simplified RR, we consider a simple randomization technique [9] with unary encoding and one randomization technique. The Memoized RR consists of a unary encoding technique, permanent randomization with memoization, and instantaneous randomization technique proposed by RAPPOR [10]. We follow the ProTECting [11] algorithm to implement the Memoized RR. The Bloom RR is similar to Memoized RR, except, we use bloom filter encoding instead of unary encoding.

While encoding the numerical data, we consider the histogram representation with 16 bins as shown in Fig. 3. For unary encoding, if data is 50, the first index will be 1 and the

TABLE I: Overview of selected privacy mechanisms for comparison

| Techniques Category | Comparisons | Evaluation |
|---|---|---|
| LDP: Distribution-based | Laplace distribution<br>vs<br>Gaussian distribution<br>vs<br>Exponential distribution<br>vs<br>Gamma distribution | MAPE: Average error among original streaming data and noisy streaming data |
| LDP: Randomized response-based | Bloom filter based RAPPOR (Bloom)<br>vs<br>Unary encoding based RAPPOR (Memoized)<br>vs<br>Simple one randomization based RAPPOR (Simplified) | MAPE: Average error among original frequency and estimated frequency |
| LDP: Hash-based | Original<br>vs<br>Count sketch-based | MAE: Average error among original relative frequency and estimated relative frequency |
| | Original<br>vs<br>Jenson-Lindenstrauss Lemma-based | Relative Frequency: Histogram of original relative frequency and estimated relative frequency |
| DP and LDP | DP<br>vs<br>LDP (Laplace)<br>vs<br>LDP (Randomized Response: Simplified) | MAPE: Average error among original frequency and estimated frequency |

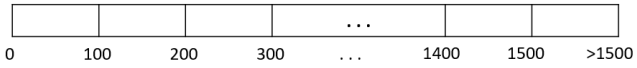



Fig. 3: Histogram bins for encoding

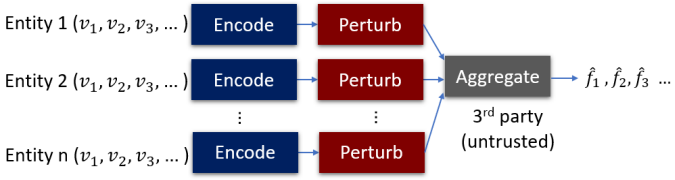



Fig. 4: Randomized response

other index value will be 0. If data is 2550, the last index will be 1.

An untrusted third party aggregates the perturbed data from different entities at every timestep and report estimated frequency (Fig. 4). The limitation is that we need multiple entity data. Higher the number of entity, we obtain higher accuracy. However, we get very low accuracy in calculating the estimated frequency for a single entity.

*3) Hash based techniques:* We consider two hash-based techniques. The count-sketch technique [12] and the Johnson Lindenstrauss random projection technique [13].

In the count-sketch technique, we consider the whole month's power consumption data and distributed them into a matrix. We then compress this original load matrix $A \in R^{(m \times n)}$ using a sketch matrix $S \in R^{(n \times s)}$. The resultant matrix will be C = AS where $C \in R^{(m \times s)}$. For the original load matrix, we consider $m$ as the number of days in a specific month and $n$ be the number of loads produced in one day. In our case, n = 144 (i.e., 10 minutes interval data). We vary the size of the sketch matrix column $s$. Lower the size of $s$ contributes to the more compressed output. The privacy parameter $\epsilon$ is calculated following Li et al. [12] . Finally, we compute the l2-norm from the resultant matrix $C$ for each day and find the normalized distribution. The advantage of count-sketch is that the communication cost is reduced based on the size of the sketch matrix. One disadvantage is that we can not compute single-point perturbation.

We also consider Johnson Lindenstrauss's random projection technique for streaming data. Each household or entity encode numerical load data as one of the $k$ categorical attributes using a histogram where $k = 16$ (Fig. 3). We then follow Bassily and Smith [13] algorithm, and it returns $k$ frequency estimates for a specific time. For streaming data, we can capture frequency estimates for every timestamp or a specific period (i.e., one day). In our case, the number of attribute $k$ can be smaller than the number of households.

### B. Our Approach: Component Variations

We have done several experiments to understand the impact of different components using the distribution-based noise technique to protect instantaneous load reporting.

First, we consider the most simple case showing in Fig. 2. Streaming values pass through the perturber and noisy streaming values will be produced. For a specific peak value $B$ (i.e., threshold or sensitivity), there are two options for the remainder load ($v_i - B$) when the current load($v_i$) is greater than the peak value. We can truncate the remainder load (i.e. without carry-on) or add the remaining load with the next streaming value $v_{i+1}$ (i.e., with carry-on).

TABLE II: Overview of component variation

| Component | Comparisons |
|---|---|
| Smoothing | No smoothing (Noisy) vs Average smoothing vs Median smoothing |
| Truncated Load | With carry-on vs Without carry-on |
| Peak value | Same peak value vs Time varying peak value |
| Noise combination | One distribution-based noise vs Sequential composition of two noises |

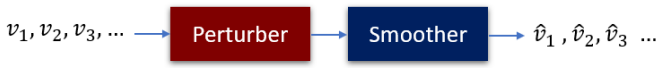



Fig. 5: Data stream perturbation with smoothing

Second, we consider the smoothing component after the perturbation as shown in Fig. 5. Several recent works use smoothing after using distribution-based noise [14], [15], [16]. We consider median smoothing and average smoothing. The smoothing from timestep $t$ is done using the perturbed data from timesteps $t$, $t_{i-1}$ and $t_{i+1}$.

Third, we consider two perturbers instead of one. For instance, we consider combining both Laplace and Exponential distribution-based noise instead of just using Laplace distribution-based noise. We combine the noises following the sequential composition property (lemma 1).

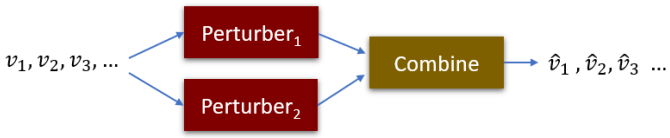



Fig. 6: Sequential composition of multiple distribution-based noises

## IV. Result

In this section, we show the evaluation. Every experiment is repeated five times and the mean value is reported. A lower privacy parameter value ($\epsilon$) represents a higher privacy guarantee with lower utility. A higher $\epsilon$ value represents a lower privacy guarantee with higher utility.

### A. Experimental Setup

We selected NREL DATA [8] for our experiments. This dataset contains randomly selected 200 households from the midwest region of the USA. It contains residential electricity demand in every 10-minute interval for the whole year of 2010.

We implement the experiments in Python 3.8 with numpy, pandas, mmh3, math, scipy, and sklearn libraries. We conduct experiments on a PC with Intel Core i7-8550U CPU and 16GB RAM. All experiments are repeated 5 times.

### B. Experimental Metrics

We consider several metrics to evaluate our experiments. Here, $x$ is denoted as the expected value and $y$ is denoted as the observed value.

We compute the mean absolute percentage error (MAPE) shown in Equation 3 for two cases. In the first case, we used to calculate the histogram comparison between different bins. In this case, we consider n as bin size. Also, we calculate the discrepancy between every streaming data point. In that case, we consider $n$ as the window size of streaming data.

$$MAPE = \frac{\sum_{i=1}^{n} |\frac{y_i - x_i}{x_i}|}{n} \tag{3}$$

We compute the relative error (Equation 4) in the cases where we have the total noisy value and actual value of a specific period.

$$Relative\ Error = |\frac{y_i - x_i}{x_i}| * 100\% \tag{4}$$

We compute mean absolute error (MAE) shown in Equation 5 for relative frequency data.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{5}$$

Mutual information measures the statistical data utilities. Mutual information between $x$ and $y$ is:

$$Mutual\ Information = \sum_{x,z} p(x,z)\ log \frac{p(x,y)}{p(x)p(y)} \tag{6}$$

Jensen-Shannon (JS) divergence measures similarity between two distribution X and Y. Lower JS divergence value denotes higher similarity.

$$JS\ Divergence = \frac{1}{2} KL(x||m) + \frac{1}{2} KL(y||m) \tag{7}$$

where $m = \frac{1}{2}(x + y)$ and $KL()$ denotes Kullback-Leibler divergence [17].

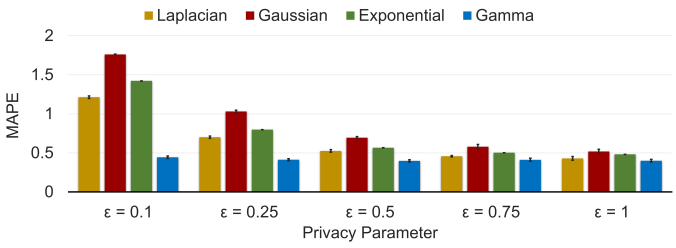



Fig. 7: Comparison of different distribution based noise varying privacy parameter

### C. Comparison of Different Distribution-based Noise

We evaluate different distribution-based techniques. The description of the approach is depicted in Section III-A1. In Fig. 7, we show the impact of different noise-base distributions varying the privacy parameter epsilon. Gamma distribution shows comparatively lower relative error among them, and Gaussian distribution shows a higher relative error.

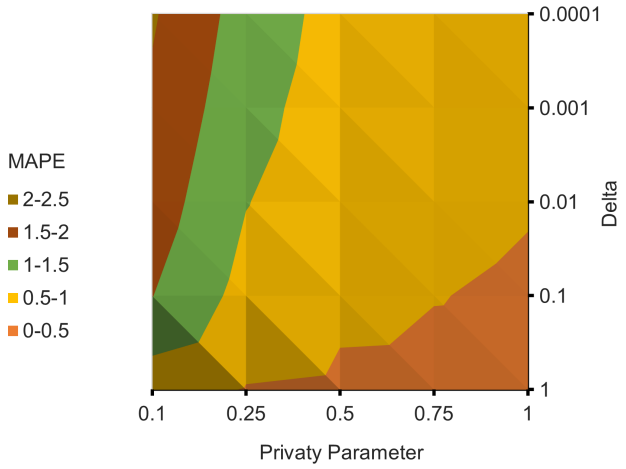



Fig. 8: Gaussian noise label varying delta and privacy parameter

The Gaussian mechanism satisfies ($\epsilon$, $\delta$)-differential privacy. Changing the value of $\delta$ and $\epsilon$ changes the error level as well (Fig. 8). We observe that higher value of $\delta$ results in higher utility and lower value of $\delta$ results in higher level of privacy.

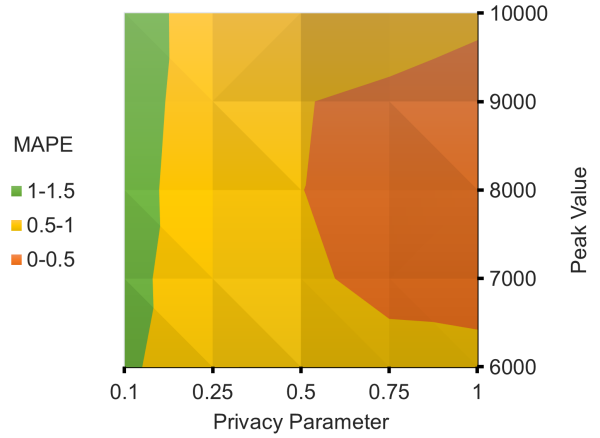



Fig. 9: Laplace noise label varying Peak value and privacy parameter

Choosing the optimal peak value (e.g., cutoff) is also an important factor. We observe that higher peak value results in higher error as the higher sensitivity introduces a high amount of noise. On the other hand, lower peak value also results in higher error as noisy load value are cut off at the peak point and the remainder is transmitted to the next stream. In the NREL dataset, the maximum load value is 14,777, and the peak value 8000 is in the 99.94 percentile.

For distribution-based comparison (Figure 2), we consider peak value of 8000 for all distributions and delta value of 0.01 for Gaussian distribution.

### D. Comparison on RAPPOR Variations for Frequency Estimation

We show the comparison of three variations of RAPPOR techniques (i.e., simplified, memoized and bloom) on streaming data in Fig. 10. We show the performance based on three metrics: MAPE (Fig. 10(a)), mutual information (Fig. 10(b)), and JS divergence (Fig. 10(c)). Simplified technique shows higher utility with lower MAPE, higher mutual information and lower JS divergence than other techniques. The reason behind the high utility is that each data only gets perturbed for one time. On the other hand, Bloom technique shows higher privacy with higher MAPE and JS divergence value. Both Bloom and Memoized use perturbation twice. However, in bloom technique, the extra level of privacy comes from hash based bloom filter encoding.

### E. Results using Count Sketch Approach

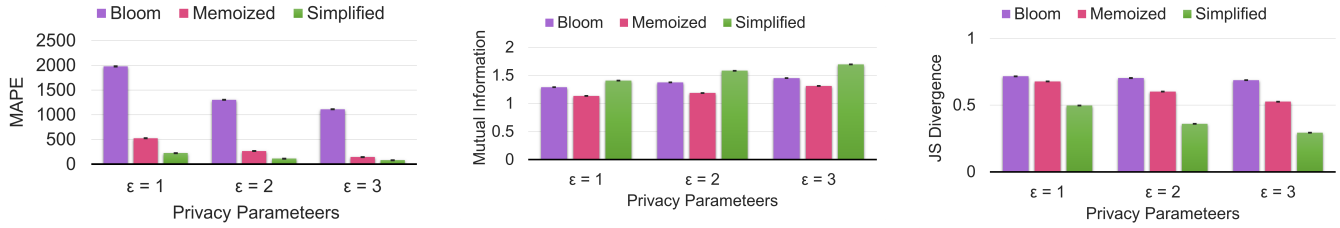
Section III-A3 describes the approach we adopt for protecting streaming data privacy using count-sketch technique. We observe that the mean absolute error and compression ratio vary with the privacy parameter (Fig. 11). In our experiment, the column of original matrix is 144. When we choose sketch matrix column 26, the compressed output matrix is 5.5 times smaller with $\epsilon$ value of 0.5 and mean absolute error (MAE) value of 0.0029. Increasing sketch matrix column results in lower privacy (i.e., higher $\epsilon$ value) and higher utility (i.e., lower MAE value).

### F. Results using JLRR Approach

We consider 200 households data at a specific timestamp for evaluating Johnson-Lindenstrauss Randomized response (JLRR) method. We show relative frequency estimation result in Fig. 12. When the original frequency estimate is very high, the JLRR shows a lower estimate than the original one. Among other cases, JLRR shows slightly higher estimate than the original about 67% times.

### G. Results on GDP vs LDP techniques

We compare the global differential privacy (GDP) technique and two local differential privacy techniques, i.e., distribution-based (laplace), randomized response based (simplified) in Fig. 13. For the GDP, we use Diffprivlib library from IBM. In this case, we consider the data lower than 75 percentile and ignored the outliers. In GDP, we compute the original frequency from 200 households and add noise before release. Therefore, we observe that the GDP shows higher utility with very low MAPE. However, a trusted aggregator is needed for GDP technique. In LDP techniques, 200 households add noise (distribution-based noise or perturbation noise) before sending their data to an aggregator who estimate the frequency.




(a) MAPE (b) Mutual Information (c) Johnson Shannon Divergence

Fig. 10: Evalution of the RAPPOR techniques

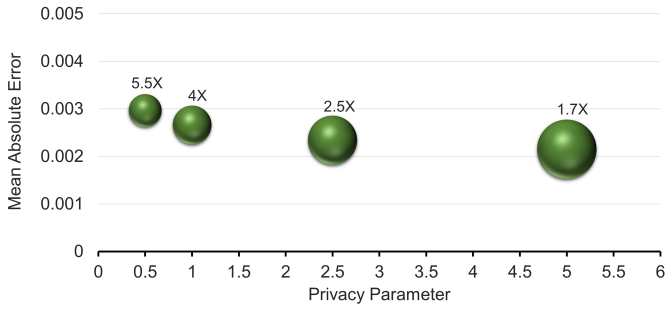



Fig. 11: Evaluation of sketch based techniques varying privacy parameters

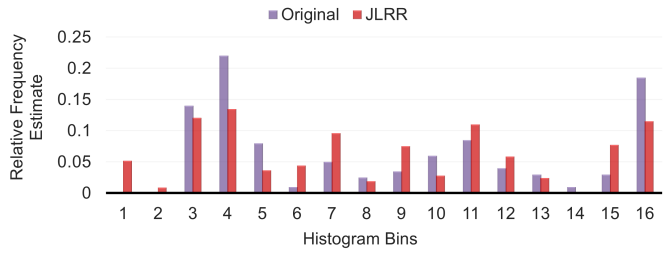



Fig. 12: Evaluation of Johnson Lindenstrauss randomized response (JLRR) approach

Randomized response-based LDP incurs higher error than distribution-based LDP.

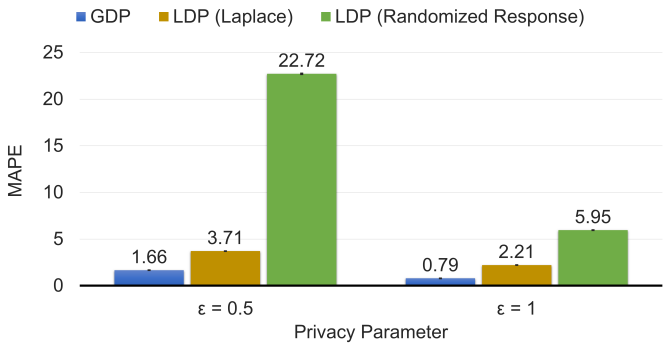



Fig. 13: Comparison of global differential privacy (GDP) and local differential privacy (LDP) techniques varying privacy parameter

### *H. Impact of varying components*

We consider instantaneous load reporting for one household for one month with load values generating in 10 minutes time intervals. We protect the specific event (i.e., higher load value) of the household using the optimal peak value with Laplace distribution-based technique. If load at a specific time is higher than a chosen peak value, then consider sending load value as peak value and add the noise depending on sensitivity. The remainder of the load can be either added to the next reading (i.e., w/ carry on) or the remainder is simply ignored (w/o carry on).

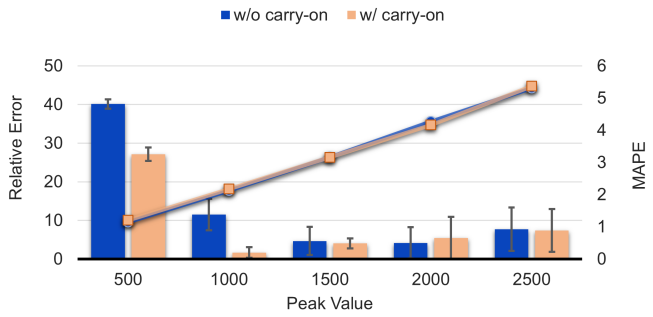



Fig. 14: Results on varying peak value. "w/o carry-on" denotes without carry-on and "w/ carry-on" denotes with carry-on truncated load value in next streaming data

We vary peak value and examine the impact on w/ carry-on and w/o carry-on in Fig. 14. The bars show relative error (error between the original sum of load and the noisy sum of load for the first month) and lines show MAPE (average error among noisy load and the original load of every timestamp). We observe the best lowest error relative error of 1.67 when the peak is 1000 for w/ carry. For the w/o carry option, the best relative error is 4.19 when the peak is 2000. Lower peak value contributes to additional loss of load for w/o carry-on option. Therefore, the relative error is much higher for w/o carry-on than w/carry-on. MAPE is similar for both cases for varying peak values.

Afterward, we consider different peak values for day and night and see the impact on w/ carry on and w/o carry on in Fig. 15. We observe the lowest error relative error of 2.02 when peak value at night is 250 and peak value at other time is 1500 for w/ carry. For the w/o carry option, the best relative error

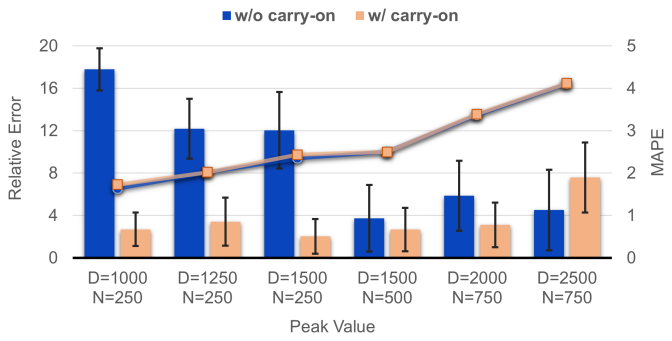



Fig. 15: Results on varying peak value at day (D) and night (N)

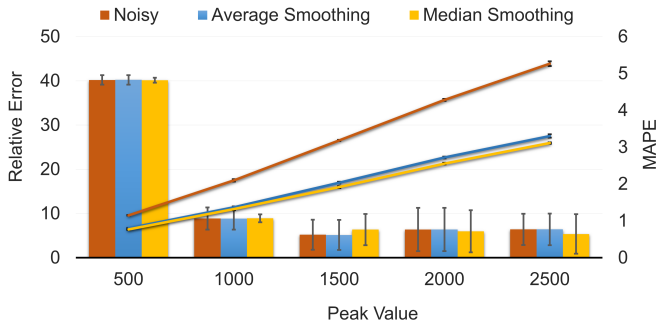



Fig. 17: Result on two smoothing techniques varying peak value

is 3.75 when the peak value at night is 500 and the peak value at other times is 1500. For w/o carry-on option, choosing a different peak value for night shows a lower error value than a single peak value. For w/ carry-on option, constant peak shows better performance. MAPE is similar for both cases for varying peaks as well.

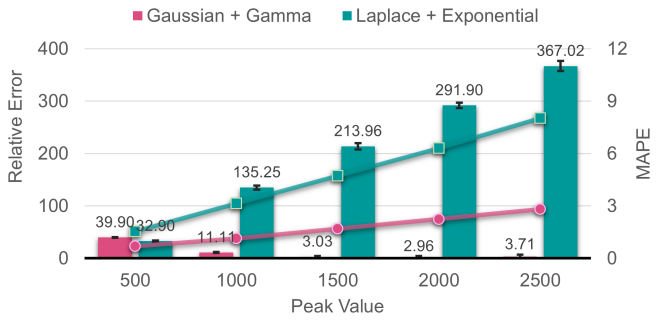



Fig. 16: Sequential composition of distribution-based noises varying peak value

We also try sequential combination of two distribution-based noises. We consider $\epsilon$ value of 0.5 for both noise distribution and use sequential combination to get 1-differential privacy. Here we choose $\delta = 1$ for Gaussian distribution-based noise. We observe that Gamma distribution and Gaussian distribution combination shows lower relative error and MAPE and Laplace and Exponential distribution combination. If user requires more privacy level, Laplace and Exponential combination can be better option. On the other hand, Gamma and Gaussian combination provides more utility.

We also explore average smoothing, median smoothing and no smoothing (i.e., noisy) in Fig. 17. We did not observe any significant difference in terms of relative error among these smoothing techniques. However, we see difference in terms of MAPE. Both average and Median smoothing shows significantly lower MAPE value. Among them, median smoothing shows slightly lower MAPE value than average smoothing.

## V. Related work

Several research works have been proposed using differential privacy and local differential privacy in academia and industry to protect streaming data from IoT devices and other edge devices.

Thorve et al. [18] propose a Laplace-based differential privacy technique for streaming data. One year's load data was clustered based on power energy level, then Laplace noise is added in each cluster and finally, the private time-series representation of each cluster are released. Robinson et al. [19] propose CASTLEGUARD that guarantees k-anonymity, l-diversity, and differential privacy at the same time. It uses Laplace distribution-based noise for perturbation and clustering to satisfy k-anonymity.

RAPPOR [10], [20] is proposed by Google that achieve $\epsilon$-LDP when a user reports a value infinite times using randomized response technique. They consider bloom filter encoding with two rounds of randomization (e.g., permanent randomization and instantaneous randomization). Among them, permanent randomization guarantees Longitudinal Privacy. ProTECting [11] also follows two round of randomization RAPPOR technique. During encoding, they apply unary encoding instead of bloom filter encoding. They use smart meter data to estimate the frequency and show that ProTECting achieves better performance than RAPPOR. PrivApprox [21] also use a randomization technique. However, they introduce sampling at the client-side for low-latency approximation before the randomization technique and also implement transmitting answers using a proxy for anonymization and unlinkability.

Adding distribution-based noise is another technique to provide privacy to streaming data. PeGaSus [15] takes a stream data and perturb the data using Laplace noise. It also utilizes a grouper module that partitions the streaming data to apply smoothing on the perturbed data. Hassan et al. [22] propose instantaneous data reporting with peak value preservation using Laplace noise. Fang et al. [23] propose local differential private streaming (LDPS) protocol for numerical and categorical attributes. LDPS satisfy local differential privacy and sliding window-based w-event privacy. For mean estimation, Duchi's [24] method and Laplace mechanism are considered. For frequency estimation, RAPPOR [10] technique is considered.

Bassily et al. [13] propose a succinct histogram proto-

col based on a random matrix project technique following Johnson-Lindenstrauss Lemma. Count sketch [25], [26] based implementation is another technique which is used by Apple [27]. Li et al. [12] propose a DiffSketch framework that uses a hash-based sketch matrix to reduce communication cost with a marginal decrease in accuracy metric.

Several techniques propose an optimized threshold optimization technique for streaming data. Perrier et al. [28] consider finding a more realistic threshold (i.e., peak value) based on a 99.5 percentile value as a threshold from a time lag. They also consider a binary tree algorithm to reduce the scale of the noise level. Wang et al. [16] propose to release a real-time data stream under differential privacy (ToPS) and local differential privacy (ToPL). They also formulate an Exponential Mechanism-based optimization algorithm to choose an optimal threshold.

The existing works on distribution-based noise mainly focus on using one type of noise (e.g., Laplace). In this paper, we show that combining more noise provides the entity (e.g., user) and third party some flexibility in determining how much noise they want in the noisy streaming data.

## VI. Conclusion

We present various privacy-preserving local differential privacy algorithms for streaming data. We compare these techniques and show their limitations and benefits. To get frequency estimation, if an entity agrees to release only aggregated streaming data (e.g., streaming data generated in one month), the count sketch-based technique is an excellent choice due to lower communication costs. If an entity agrees to release streaming data in every timestep, the bloom filter-based RAPPOR technique provides a higher privacy guarantee.

We also vary different components for distribution-based noise for instant noisy reporting of the streaming data and present when they can be useful. The smoothing technique after the perturbation is beneficial if the third party is interested in examining every timestamp noisy streaming data. Combining different noise-based techniques also offers a wide range of options for data privacy-utility tradeoff.

As future work, we plan to show demonstrations of local differential privacy techniques from physical devices. We are also interested in using perturbed data in different machine learning algorithms.

## VII. Acknowledgement

Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U.S. Department of Energy under contract DE-AC05-00OR22725.

## References

[1] C. Dwork, A. Smith, T. Steinke, and J. Ullman, "Exposed! a survey of attacks on private data," *Annual Review of Statistics and Its Application*, vol. 4, pp. 61–84, 2017.

[2] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Knock knock, who's there? membership inference on aggregate location data," *arXiv preprint arXiv:1708.06145*, 2017.

[3] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, 2014, pp. 17–32.

[4] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," *arXiv preprint arXiv:2007.07646*, 2020.

[5] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*. Springer, 2008, pp. 1–19.

[6] M. Joseph, A. Roth, J. Ullman, and B. Waggoner, "Local differential privacy for evolving data," 2018.

[7] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 19–30.

[8] M. Muratori, "Impact of uncoordinated plug-in electric vehicle charging on residential power demand," *Nature Energy*, vol. 3, no. 3, pp. 193–201, 2018.

[9] J. P. Near and C. Abuah, *Programming Differential Privacy*, 2021, vol. 1. [Online]. Available: https://uvm-plaid.github.io/programming-dp/

[10] U. 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*, ser. CCS '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1054–1067. [Online]. Available: https://doi.org/10.1145/2660267.2660348

[11] I. de Castro Vidal, A. L. da Costa Mendonça, F. Rousseau, and J. de Castro Machado, "Protecting: An application of local differential privacy for iot at the edge in smart home scenarios," in *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC, 2020, pp. 547–560.

[12] T. Li, Z. Liu, V. Sekar, and V. Smith, "Privacy for free: Communication-efficient learning with differential privacy using sketches," *CoRR*, vol. abs/1911.00972, 2019. [Online]. Available: http://arxiv.org/abs/1911.00972

[13] R. Bassily and A. Smith, "Local, private, efficient protocols for succinct histograms," in *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, ser. STOC '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 127–135. [Online]. Available: https://doi.org/10.1145/2746539.2746632

[14] G. Eibl and D. Engel, "Differential privacy for real smart metering data," *Comput. Sci.*, vol. 32, no. 1–2, p. 173–182, Mar. 2017. [Online]. Available: https://doi.org/10.1007/s00450-016-0310-y

[15] Y. Chen, A. Machanavajjhala, M. Hay, and G. Miklau, *PeGaSus: Data-Adaptive Differentially Private Stream Processing*. New York, NY, USA: Association for Computing Machinery, 2017, p. 1375–1388. [Online]. Available: https://doi.org/10.1145/3133956.3134102

[16] T. Wang, J. 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," *ArXiv*, vol. abs/2005.11753, 2020.

[17] "Kullback–leibler divergence," https://en.wikipedia.org/wiki/Kullback-Leibler_divergence, accessed: August 12, 2021.

[18] S. Thorve, L. Kotut, and M. Semaan, "Privacy preserving smart meter data," in *Proceedings of The 7th International Workshop on Urban Computing (UrbComp'18)*, 2018.

[19] A. Robinson, F. Brown, N. Hall, A. Jackson, G. Kemp, and M. Leeke, "Castleguard: Anonymised data streams with guaranteed differential privacy," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2020, pp. 577–584.

[20] G. Fanti, V. Pihur, and U. Erlingsson, "Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, 03 2015.

[21] D. L. Quoc, M. Beck, P. Bhatotia, R. Chen, C. Fetzer, and T. Strufe, "Privapprox: Privacy-preserving stream analytics," in *Proceedings of the 2017 USENIX Conference on Usenix Annual Technical Conference*, ser. USENIX ATC '17. USA: USENIX Association, 2017, p. 659–672.

[22] M. U. Hassan, M. H. Rehmani, R. Kotagiri, J. Zhang, and J. Chen, "Differential privacy for renewable energy resources based smart metering," *Journal of Parallel and Distributed Computing*, vol. 131, pp. 69–80, 2019.

[23] X. Fang, Q. Zeng, and G. Yang, "Local differential privacy for data streams," in *Security and Privacy in Digital Economy*, S. Yu, P. Mueller, and J. Qian, Eds. Singapore: Springer Singapore, 2020, pp. 143–160.
[24] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Minimax optimal procedures for locally private estimation," *Journal of the American Statistical Association*, vol. 113, no. 521, pp. 182–201, 2018.
[25] J. Upadhyay, "Differentially private linear algebra in the streaming model," *arXiv preprint arXiv:1409.5414*, 2014.
[26] "Count sketch," http://wangshusen.github.io/code/countsketch.html, accessed: August 12, 2021.
[27] "Learning with privacy at scale," https://machinelearning.apple.com/research/learning-with-privacy-at-scale, accessed: August 12, 2021.
[28] V. Perrier, H. J. Asghar, and D. Kaafar, "Private continual release of real-valued data streams," *arXiv preprint arXiv:1811.03197*, 2018.