Analyzing Data Privacy for Edge Systems

1st Olivera Kotevska
Oak Ridge National Laboratory
Oak Ridge, TN, USA
kotevskao@ornl.gov

2nd Jordan Johnson
Oak Ridge National Laboratory
Oak Ridge, TN, USA
johnsonja1@ornl.gov

3rd A. Gilad Kusne
National Institute of Standards and Technology
Gaithersburg, MD, USA
aaron.kusne@nist.gov

***Abstract*—Internet-of-Things (IoT)-based streaming applications are all around us. Currently, we are transitioning from IoT processing being performed on the cloud to the edge. The increasing number of deployed IoT systems generates an ever-growing volume of data, making edge processing more desirable. While moving to the edge provides significant networking efficiency benefits, IoT edge computing creates significant data privacy concerns.**

We propose a methodology that can successfully privacy protect the continual data streams generated by sensors on the edge device. We implement local differential privacy on streaming data and incorporate Bayesian inference and Gaussian process to test for privacy policy vulnerability. We demonstrate our methodology on a real-world smart meter testbed and identify the optimal privacy protection settings.

***Index Terms*—privacy, edge, Bayesian, algorithms, streaming data, IoT**

I. Introduction

We live in the era of Internet-of-Things (IoT). Their benefits are well understood across a wide range of applications including healthcare monitoring [29], smart homes [39], and grid systems [17]. Still, their vulnerabilities and the importance of data privacy are not taken seriously [26], [41].

IoT has become a primary target for cyberattacks, and the repeated security incidents on IoT devices represents a rising trend. The rapid increase of connected devices has created multiple targets for attackers (e.g., security cameras, smart TVs, connected printers, smart bulbs, coffee machines, internet-connected gas stations). Furthermore, recent scandals of user monitoring through Roku TV and Amazon Fire TV [31] demonstrate avenues for privacy breach of user personal information [1]. While the benefit of the vast amount of data availability is undeniable [20], a better mechanism for privacy protections is needed.

Usually, security practices are established with passwords and access point permissions. But, the need to secure data storage, transfer and communicate securely is mainly addressed by cryptography and other formal approaches [43]. Although cryptography offers security, it requires a significant amount of computation which makes it challenging for IoT-based systems with limited resources to process encrypted data efficiently. Additionally, there is a risk that unauthorized users could gain access to the data after it has been decrypted, giving them full access to the sensitive details [37]. Privacy algorithms provide a solution for overcoming these challenges.

Firstly, to clarify, there is a difference between security and privacy. The fact that our data is securely stored today does not mean that our privacy is protected, neither today nor in the future [41]. Cryptography techniques have proven security guarantees, and while security is established, these techniques do not necessarily guarantee privacy. Cryptography's weakness is that it does not protect endless data streams or as the data arrives. In contrast, privacy approaches provide mechanisms for protecting the sensitive information within the data, such as personal protected information (PII), as soon as the data is generated and without a key to discover the original raw data.

In the context of IoT-based systems, the data is constantly streaming. The aim is to privacy protect the streaming data near the source and before it is sent to the cloud. This approach requires processing on the edge device, and we need privacy algorithms that do not require high processing capabilities [30]. When this privacy mechanism is implemented, it will increase the trust in the systems and enable more users to use them with confidence.

This work investigates privacy algorithms for streaming applications and their performance on the edge-based system. Mainly the focus is on differential privacy (DP) algorithms [22]. An algorithm is DP if an observer cannot tell if an individual's sensitive information was used in the computation. One system is considered differential private if when sharing information about the dataset, it describes the patterns within the dataset while withholding sensitive information about individuals or entities in the dataset.

This work looked into privacy techniques used for streaming applications such as IoT. We experimented with different privacy algorithms and demonstrated their performance on an actual testbed using various streaming settings. We explore the solutions based on local differential privacy [15] (e.g., distribution-based techniques) and design their adaptation for streaming data. We developed a Bayesian inference algorithm for streaming data to measure the uncertainty bounds of local differential privacy parameters. Finally, we compare the methods and let the interested party know which methods are beneficial for them to use.

We show that combining the distribution-based noise approach with Bayesian inference widens the privacy assurance window. We examine the performances for each technique

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The publisher acknowledges the US government license to provide public access under the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

under different privacy settings and the uncertainty bounds when only a few samples are available. We also vary different data frequencies and time windows and present when they are the most useful.

The outline of the paper is organized as follows. First, we describe the streaming algorithms in Section III-C and edge systems in Section II. We explain the testbed and present the results in Section IV-D3. We report the existing related works in Section V. Finally we conclude in Section VI.

II. Edge Computing

Edge computing is a computing pattern that brings data processing closer to the source or where it is created [12]. It is a decentralized topology based on keeping data local, at the *edge* of the network, as close to the source as possible. Using the edge directly on or near the source increases the efficiency and speed of data use and reduces unnecessary network burden and data traffic waste. The aim is to bring cloud capabilities closer to the user.

Similar paradigms are fog computing and cloudlets, and the difference is where the computing power and intelligence power are placed. In architecture settings, the edge is between the cloud and devices. The edge collects the data from local sensors and performs an analysis. Depending on how computing-intensive the tasks are, the edge can be categorized as micro, thin, and thick [6].

Edge computing offers a more efficient approach to processing data and not overloading the cloud with all processing steps. In cases when data needs to be transferred over the network, privacy guarantee is necessary to ensure trustful data transfer. Integrating privacy algorithms at the edge can provide confidence in protecting sensitive information and using edge-based systems.

III. Algorithms

A. Data Streaming Characteristics

Streaming data is generated by video platforms (e.g., Netflix), music platforms (e.g., Spotify), intelligent virtual assistants (e.g., Alexa), and any IoT-based system (e.g., Smart Things). The data type can be video, audio, text, or numerical format. It is characterized by continuous data generation, dynamic evolving data, heterogeneous data types, and near real-time processing.

B. Local Differential Privacy

The goal of privacy algorithms is to protect the sensitive information in the dataset and guarantee the desired level of privacy.

In *Local Differential Privacy (LDP)* [15], the data is perturbed first before sending it to an aggregator for analysis (Fig. 1). The advantage of LDP is that there is no need for a trusted data aggregator.

Definition: A randomized mechanism F guarantees ϵ-*LDP* ($\epsilon \geq 0$) for any pair of input values v and $v' \in S$ if and only if F satisfies:

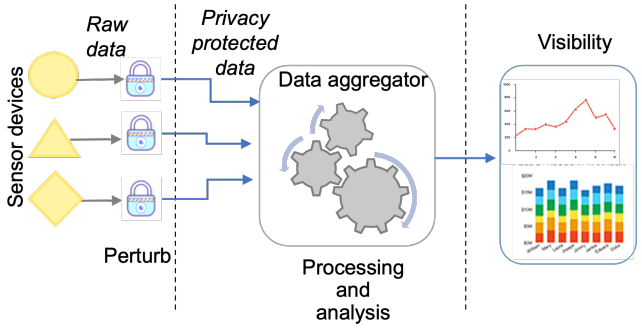

Fig. 1. Local differential privacy (LDP)

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

Here, O is the subset of output.

There are a few algorithm types for LDP [13]. We focus on distribution-based techniques and test their performance on an edge system. We apply four well-known distribution-based noise mechanisms (e.g., Laplace, Gaussian, Exponential, and Gamma).

The *Laplace distribution* with the probability distribution function [8]:

$$p(x) = \frac{1}{2b} \exp(-\frac{|x-\mu|}{b}) \tag{2}$$

where, b is the scale and μ the mean.

The *Gaussian distribution* with the probability distribution function [4]:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{3}$$

where, σ is the scale parameter and μ is the mean.

The *Exponential distribution* with the probability distribution function [2]:

$$p(x) = \frac{1}{b} \exp(-\frac{x}{b}) \tag{4}$$

The *Gamma distribution* with the probability distribution function [3]:

$$p(x) = x^{k-1} \frac{exp(-\frac{x}{\theta})}{\theta^k \Gamma(k)} \tag{5}$$

where, k is the shape, θ is the scale and Γ is the Gamma function.

In our case, the raw data of every reading is perturbed by adding random noise generated from different distributions. In the experiments, the scale of the noise (e.g., sensitivity) is determined by the maximum allowed noise from the utility (i.e., lower MAE value) and user (see Eq. 8 and Eq. 9).

C. Bayesian Inference

Bayesian inference is a statistical method for determining the probability of a hypothesis through the use of Bayes' theorem. [36].

A common application for Bayesian inference is using observed data Y to determine the parameter values of the data

generating model $M(\theta)$, where θ is the set of model parameters. Here we wish to infer likely values for the parameters θ. This challenge can be solved by employing Bayes' theorem. Prior belief for the value of the parameters θ is given by the probability density function $p(\theta)$ (or the probability mass function $P(\theta)$ if the parameters θ take on discrete values.) The probability of observing data Y given particular values for the parameters is given by $p(Y|M(\theta))$. Through the use of Bayes' theorem, this probability combined with the prior over θ can be combined to determine the probability of different values of θ given the observed Y. This probability is also called the posterior and is represented by $p(M(\theta)|Y)$.

Bayesian inference allows one to employ prior knowledge over possible values of the parameters θ and prior knowledge of the relationship between the observed variable and model (given by $p(Y|M(\theta))$) to statistically infer a distribution over the unknown parameters.

In this work, we use Bayesian inference to determine the accuracy with which a malevolent actor can identify the privacy policy parameters employed from a compromised data stream. Then we investigate how well the actor can utilize this information to estimate originating raw data for other privacy protected data.

IV. Demonstration

A. Our Testbed

For our testbed, we decided to focus on a power systems example. Traditionally, power flows from large power generation facilities down to consumers. However, with the rise of the smart grid [23] and renewable energy technologies, the flow of power has become a two-way exchange. This means the grid must be able to adapt to not only variable power needs but also variable power production from the top and bottom. This is being solved with smart grid technologies such as a two-way communication infrastructure that allows utility companies to collect real-time data from customers.

One example of a two-way communication is advanced metering infrastructure (AMI) described in [32]. AMI includes a number of technologies, one of which is the collection of data from smart meters attached to homes and buildings. These smart meters monitor real-time power usage data that can be retrieved by the utility company as often as desired. This presents a privacy issue as others have shown that energy data can reveal sensitive information about residents and their daily activities [33].

As such, we designed our test bed to include a smart meter attached to a group of solar panels. The meter records power generation of the solar panels when it is being consumed by the local power grid, as power is generated on demand. The power usage data is retrieved by a controller at a regular, configurable interval which records it for operators to view from a human-machine interface (HMI). In our case, all of these devices are on a local network, but in a real world environment, a centralized monitoring station would collect the data from many controllers remotely across the Internet. This makes the controller a target for cyber attacks that could compromise the privacy of customers if the data was stolen.

To introduce a privacy protection algorithm, we installed a Raspberry Pi with two network interfaces in between the meter and the controller as shown in Fig. 2. Requests from the controller are forwarded to the meter, while responses are parsed by the proxy device and passed to the algorithm. The result is reformed into a response packet and sent to the controller. Because the proxy device is the only direct connection to the smart meter, all traffic is handled appropriately, and the total amount of traffic is low enough that a Raspberry Pi is easily capable of processing it within a manageable amount of time as discussed in Section IV-D.

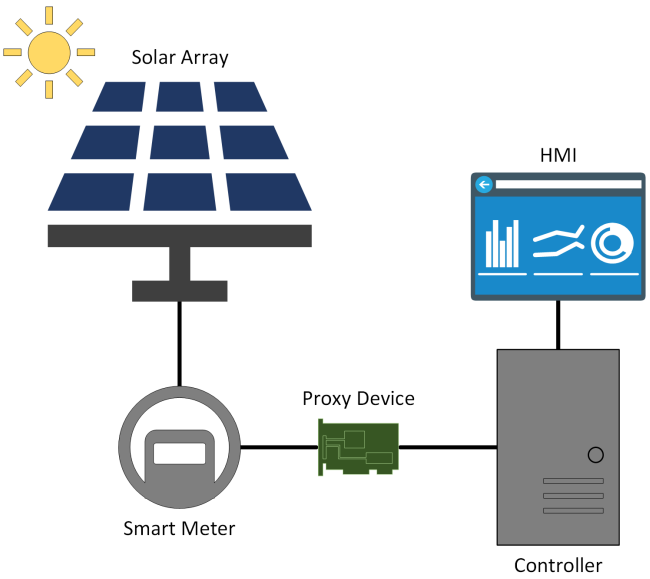

Fig. 2. A Raspberry Pi acts as a proxy device in between the meter and the controller to perform the privacy algorithm before responses are sent to the controller.

B. Dataset Description

The dataset generated by our testbed represents the power being generated by the solar panels measured in watts at that moment of time.

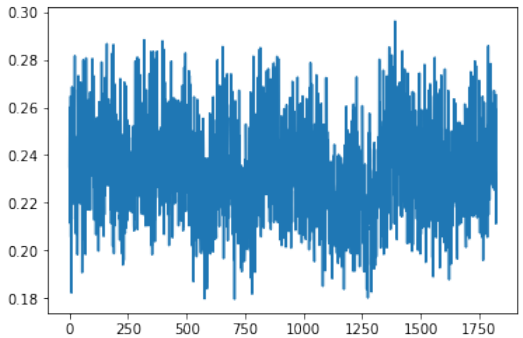

Fig. 3. Original raw smart meter data measured per second.

C. Metrics

Metrics for LDP are grouped as *error-based* and *information-theoretical* metrics [42].

Error-based metrics describe the error between the private observation and the original (real) observation. Typically, mean absolute error is used (see Eq. 6). Here, x is denoted as the expected original value, y is denoted as the observed private value and n is the total number of samples.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{6}$$

Information-theoretical metrics are adopted to quantify the error between the original data and private data. Typically, Kullback–Leibler divergence is used (see Eq. 7). The Kullback–Leibler divergence (KL divergence) is commonly adopted to measure the similarity between distributions. The KL divergence between X and Z is

$$D_{kl}(P||Q) = \int_{-\infty}^{\infty} p(x) log(\frac{p(x)}{q(x)}) dx \tag{7}$$

where P and Q denote the original distribution and private distribution [7].

D. Experiments

Privacy algorithms were implemented using Python 3.7 with standard python packages [11] and the SciPy [10] library. Bayesian inference was implemented with Python 3.7 and with the Pyro package [9] using the NUTS solver and 5,000 steps. Gaussian process was implemented using GPy library [5]. All experiments are repeated 5 times and the average reported.

The methodology in Fig. 4 shows the flexibility of choosing user preferred settings we have developed.

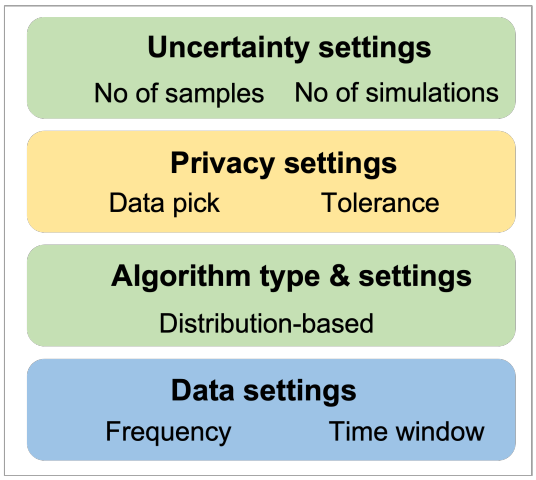

Fig. 4. Privacy framework for streaming data on the edge.

1) Measuring the Impact of Algorithm Type: We evaluate the distribution-based techniques to measure their performance for the edge device. In Fig. 5, we present the impact of different noise-base distributions varying the privacy loss. Results show that Gamma distribution shows comparatively lower relative error, and Exponential distribution shows a higher relative error under different ϵ. This was shown by using the KL metric with Gamma being close to zero and other distributions between ten and thousand.

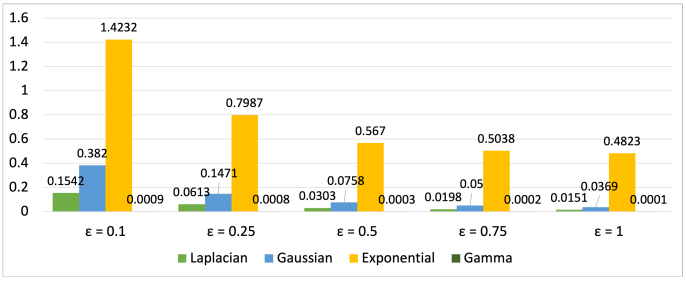

Fig. 5. Comparison of different distribution based noise varying privacy parameter.

When we look closer into the results per distribution, it shows that changing the value of δ and ϵ changes the error level as well, as expected due to the inverse relationship between the parameter values and the generating additive noise standard deviation σ, see Figure 6.

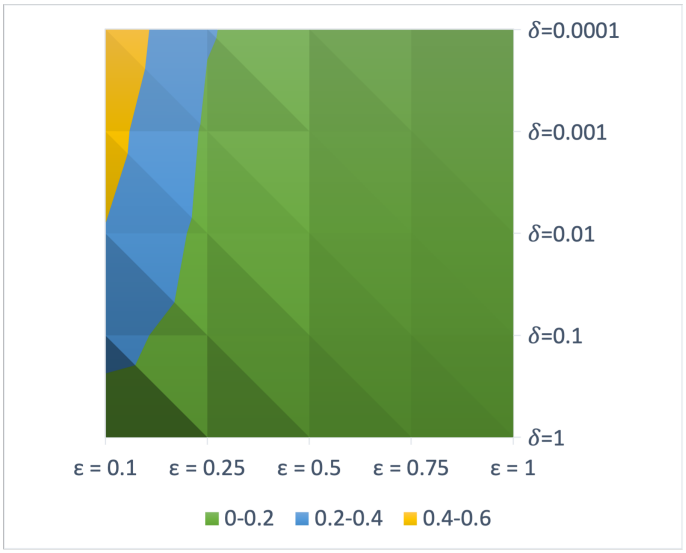

Fig. 6. Gaussian noise label varying delta and epsilon.

We observe the expected result that a higher value of δ results in higher utility (lower MAE), and a lower value of δ results in a higher level of privacy.

2) Measuring the Impact of Streaming Properties: We performed two sets of experiments based on data streaming properties.

a) Impact of data collection frequency These experiments evaluate the difference in performance when configuring the rate at which data is collected from the smart meter. We tested what we determined to be reasonable collection frequencies of 1, 5, 10, 15, and 20 seconds.

In our settings, we noticed that the data frequency impacts the algorithm performance (see Fig. 7). Results show that

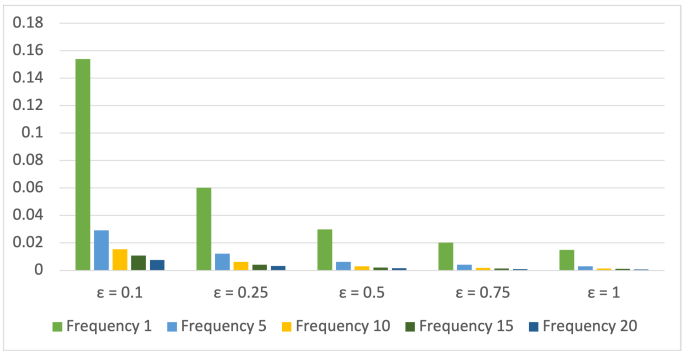

Fig. 7. Evaluation of Laplacian approach with different frequencies.

when data is collected every second, the relative error is highest, while for the other cases is significantly lower.

b) Impact of time window We ran experiments when the privacy algorithm is not processing the data as it comes, but instead collects it and processes it periodically (e.g., every 1, 5, 10, 15, 20 seconds).

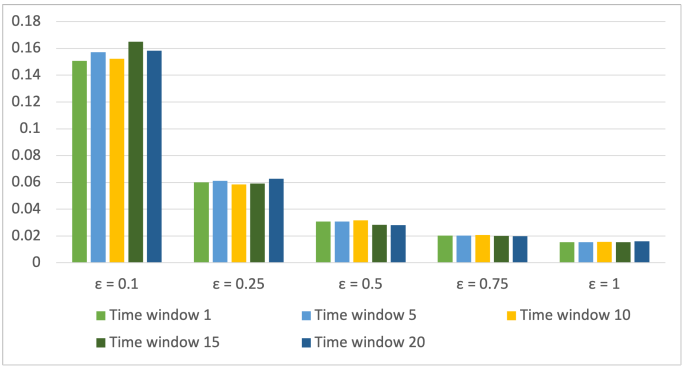

Fig. 8. Evaluation of Laplacian approach varying different time window processing.

The results show that the time window does not impact the results significantly. The time window of 5 seconds is slightly better for 0.1 and 0.25.

3) Measuring Privacy Vulnerability: We performed Bayesian inference experiments to determine the privacy policy and privacy policy parameters used for privacy protection.

We assume that the target data stream is privacy protected using the equation: $y_i = y_i + n_i$ with data index i and noise n_i given by either the Gaussian or Laplacian distribution with mean of zero and scale (or standard deviation) given by the following equations:

$$\sigma = \sqrt{2 * log(\frac{1.25}{\delta})} * \frac{sensitivity}{\varepsilon} \tag{8}$$

$$sensitivity = \sqrt{\frac{MaxAE^2}{2}} \tag{9}$$

where MaxAE is maximum allowed error calculated based on original raw data signal divided by maximum allowed noise by utility and user.

The malevolent actor is able to obtain data samples prior to privacy protection (e.g., through hacking the data source) along with the same data after privacy protection. Using Bayesian inference and analysis, the actor attempts to determine which of the two privacy policies is employed and the value of the policy parameters. Here the actor targets parameters $epsilon$ and $delta$.

a) Parameters determination: We evaluate the impact of different Bayesian inference settings and data samples on discovering the privacy parameters δ and ϵ.

The results show (see Figures 9 and 10), that as expected, with a greater number of data points, one can better identify the generating model parameter values of $epsilon$ and $delta$. As variance of the added noise is dependent on the data point intensities, it is likely that inference performance will increase as the analyzed data becomes more representative of the full range of data stream values.

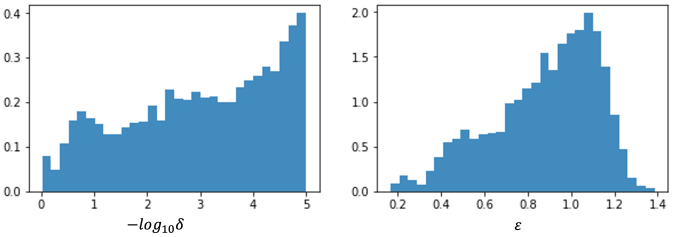

Fig. 9. Bayesian inference on Gaussian approach with 10 samples.

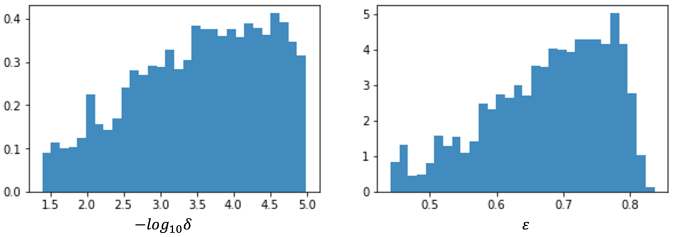

Fig. 10. Bayesian inference on Gaussian approach with 50 samples.

Using equation 8 and the expected value for ϵ and δ, computed from the posterior, we can then estimate the noise variance at each timestamp. Here we use the noisy data as an approximation for the original data in computing the sensitivity. Knowing the noise variance, we can then apply a heteroscedastic Gaussian process to estimate the value of the data prior to privacy protection for new privacy protected data. In Figure 11 the data prior to privacy protection (black dots) and the data after privacy protection (black crosses) are shown. The Gaussian process model properly identifies the trend in the originating data as shown by the Gaussian process mean (blue line).

b) Model determination: In this demonstration, model determination succeeds in both cases - when the privacy policy

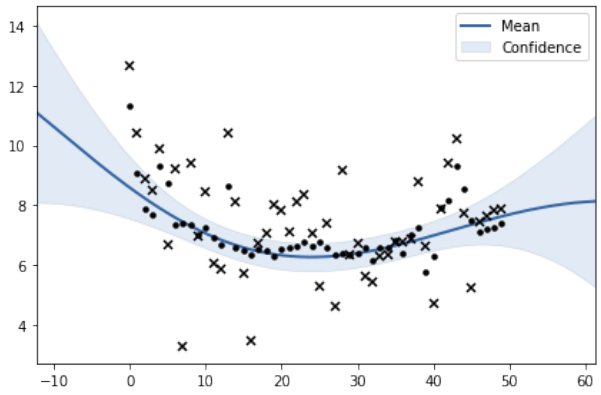

Fig. 11. Determining pattern of original data.

is Gaussian or Laplacian, as shown by the log likelihood values in Table I. Interestingly the log likelihood values are quite similar for the Gaussian and Laplacian model hypotheses when the data is generated with the Gaussian model. This suggests possible difficulty in differentiating between privacy models for additive Gaussian noise. This is positive as it serves to confound a potential malevolent actor. Success and failure may depend on the particular parameter values chosen and the range of data point values used in the study.

Data/Model	Gaussian data	Laplacian data
Gaussian model	-5.29	-4.83
Laplacian model	-5.39	-3.78

TABLE I
COMPARISON BETWEEN BAYESIAN MODELS UNDER GAUSSIAN AND LAPLACIAN PRIVATE DATASETS.

V. Related Work

There is a lot of work for privacy in streaming applications but for the purpose of this study we focused on local differential privacy approaches. Event-level differential privacy was used by Perrier *et al*. [35], Joseph *et al*. [27]. PeGaSus [19] took a data stream and perturbed the data using Laplace noise. They utilized a grouper module that partitions the streaming data to apply smoothing on the perturbed data. Hassan *et al*. [25] proposed instantaneous data reporting with peak value preservation using Laplace noise. Fang *et al*. [24] and Nguyên *et al*. [34] proposed a local differential private streaming (LDPS) protocol for numerical and categorical attributes.

An aggregated data stream approach was proposed by Huo *et al*. [26], where they applied event-level differential privacy model based on the Laplacian distribution. While the aggregated approaches have their benefits they are not applicable to some real scenarios. In those cases sliding window approaches are more appropriate, for instance Cao *et al*. [18] explored a stream-based management system for simultaneous queries. Kellaris *et al*. [28] presented a sliding window model that combines user-level and event-level differential privacy to capture a wide range of multiple events occurring at continuous time segments.

Others works focused on using local differential privacy and execution on the edge. Bi *et al*. [16] proposed local collection method based on Voronoi grid and random disturbance mechanism. Wang *et al*. [40] designed a framework for automatically protecting the sensitive features using local differential privacy. Bao *et al*. [14] demonstrated successful use of local differential privacy on mobile edge system for voting systems before the data is sent to the cloud for processing. A similar approach was presented by Usman *et al*. [38] to preserve the privacy of end-devices. However, they do not have deployment on a real-world testbed, which we have demonstrated.

Closest to our application system is the work by Dong *et al*. [21] who developed a differential private model based on Laplacian noise for solar generators. However, they used only the usual error measurement metric while we incorporate Bayesian inference to show the strength of the privacy algorithms. We have not found a work that uses Bayesian inference to characterize local differential privacy algorithms.

VI. Conclusion and Future Work

The importance of data privacy integration for internet-connected systems is clear and is slowly becoming an integral part of system software. In this work, we presented algorithms that can be used for privacy protecting streaming data on the edge and demonstrated their capabilities in real-world testbed settings. Results show that data frequency impacts the results while time window is more linear across the techniques. We also demonstrated the use of Bayesian inference to determine the venerability of the distribution techniques. We show that Bayesian inference can be used identify privacy policies that are more resilient to cyber attack.

In the future we aim to integrate local differential privacy in federated machine learning algorithms and demonstrate functionality on the edge system. We aim to also extend the capabilities of the described methodology to support a wider range of privacy algorithms and data types.

Acknowledgments

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] 533 million facebook users' phone numbers and personal data have been leaked online. https://www.businessinsider.com/stolen-data-of-533-million-facebook-users-leaked-online-2021-4. Accessed: February 15, 2022.
[2] Exponential distribution. https://numpy.org/doc/stable/reference/random/generated/numpy.random.exponential.html. Accessed: March 10, 2022.
[3] Gamma distribution. https://numpy.org/doc/stable/reference/random/generated/numpy.random.gamma.html. Accessed: March 10, 2022.
[4] Gaussian distribution. https://numpy.org/doc/stable/reference/random/generated/numpy.random.normal.html. Accessed: March 10, 2022.
[5] Gpy library. https://sheffieldml.github.io/GPy/. Accessed: March 10, 2022.
[6] Iot edge computing – what it is and how it is becoming more intelligent. https://iot-analytics.com/iot-edge-computing-what-it-is-and-how-it-is-becoming-more-intelligent/. Accessed: February 16, 2022.

[7] Kullback–leibler divergence. http://hanj.cs.illinois.edu/cs412/bk3/KL-divergence.pdf. Accessed: August 12, 2021.

[8] Laplacian distribution. https://numpy.org/doc/stable/reference/random/generated/numpy.random.laplace.html. Accessed: March 10, 2022.

[9] Pyro library. https://pyro.ai/examples/index.html. Accessed: March 10, 2022.

[10] Scipy library. https://scipy.org. Accessed: March 10, 2022.

[11] Standard python library. https://docs.python.org/3/library/. Accessed: March 10, 2022.

[12] What is edge computing. https://www.ibm.com/cloud/what-is-edge-computing. Accessed: February 16, 2022.

[13] Sharmin Afrose, Danfeng Daphne Yao, and Olivera Kotevska. Measurement of local differential privacy techniques for iot-based streaming data. In *2021 18th International Conference on Privacy, Security and Trust (PST)*, pages 1–10, 2021.

[14] Ting Bao, Lei Xu, Liehuang Zhu, Lihong Wang, Ruiguang Li, and Tielei Li. Privacy-preserving collaborative filtering algorithm based on local differential privacy. *China Communications*, 18(11):42–60, 2021.

[15] Björn Bebensee. Local differential privacy: a tutorial. *arXiv preprint arXiv:1907.11908*, 2019.

[16] Mengnan Bi, Yingjie Wang, Zhipeng Cai, and Xiangrong Tong. A privacy-preserving mechanism based on local differential privacy in edge computing. *China Communications*, 17(9):50–65, 2020.

[17] Hui Cao, Shubo Liu, Longfei Wu, Zhitao Guan, and Xiaojiang Du. Achieving differential privacy against non-intrusive load monitoring in smart grid: A fog computing approach. *Concurrency and Computation: Practice and Experience*, 31(22):e4528, 2019.

[18] Jianneng Cao, Qian Xiao, Gabriel Ghinita, Ninghui Li, Elisa Bertino, and Kian-Lee Tan. Efficient and accurate strategies for differentially-private sliding window queries. In *Proceedings of the 16th international conference on extending database technology*, pages 191–202, 2013.

[19] Yan Chen, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. *PeGaSus: Data-Adaptive Differentially Private Stream Processing*, page 1375–1388. Association for Computing Machinery, New York, NY, USA, 2017.

[20] Federal Trade Commission et al. Internet of things: Privacy & security in a connected world. *Washington, DC: Federal Trade Commission*, 2015.

[21] Jin Dong, Teja Kuruganti, Seddik Djouadi, Mohammed Olama, and Yaosuo Xue. Privacy-preserving aggregation of controllable loads to compensate fluctuations in solar power. In *2018 IEEE Electronic Power Grid (eGrid)*, pages 1–5. IEEE, 2018.

[22] Cynthia Dwork and Rebecca Pottenger. Toward practicing privacy. *Journal of the American Medical Informatics Association*, 20(1):102–108, 2013.

[23] Xi Fang, Satyajayant Misra, Guoliang Xue, and Dejun Yang. Smart grid—the new and improved power grid: A survey. *IEEE communications surveys & tutorials*, 14(4):944–980, 2011.

[24] Xianjin Fang, Qingkui Zeng, and Gaoming Yang. Local differential privacy for data streams. In Shui Yu, Peter Mueller, and Jiangbo Qian, editors, *Security and Privacy in Digital Economy*, pages 143–160, Singapore, 2020. Springer Singapore.

[25] Muneeb Ul Hassan, Mubashir Husain Rehmani, Ramamohanarao Kotagiri, Jiekui Zhang, and Jinjun Chen. Differential privacy for renewable energy resources based smart metering. *Journal of Parallel and Distributed Computing*, 131:69–80, 2019.

[26] Yan Huo, Chengtao Yong, and Yanfei Lu. Re-adp: real-time data aggregation with adaptive-event differential privacy for fog computing. *Wireless Communications and Mobile Computing*, 2018, 2018.

[27] Matthew Joseph, Aaron Roth, Jonathan Ullman, and Bo Waggoner. Local differential privacy for evolving data. *arXiv preprint arXiv:1802.07128*, 2018.

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

[29] Saso Koceski, Olivera Kotevska, Elena Vlahu-Gjorgievska, and Vladimir Trajkovik. Continuous realtime monitoring of patient's vital signs based onzigbee standard. *International Journal of Informatics and Communication Technology (IJ-ICT)*, 3(2), 2014.

[30] Zhuoran Ma, Jianfeng Ma, Yinbin Miao, Ximeng Liu, Kim-Kwang Raymond Choo, Ruikang Yang, and Xiangyu Wang. Lightweight privacy-preserving medical diagnosis in edge computing. *IEEE Transactions on Services Computing*, 2020.

[31] Hooman Mohajeri Moghaddam, Gunes Acar, Ben Burgess, Arunesh Mathur, Danny Yuxing Huang, Nick Feamster, Edward W Felten, Prateek Mittal, and Arvind Narayanan. Watching you watch: The tracking ecosystem of over-the-top tv streaming devices. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 131–147, 2019.

[32] Ramyar Rashed Mohassel, Alan Fung, Farah Mohammadi, and Kaamran Raahemifar. A survey on advanced metering infrastructure. *International Journal of Electrical Power & Energy Systems*, 63:473–484, 2014.

[33] BJ Murrill, EC Liu, and RM Thompson. Smart meter data: Privacy and cybersecurity. congressional research service. In *Library of Congress*, 2012.

[34] Thông T Nguyên, Xiaokui Xiao, Yin Yang, Siu Cheung Hui, Hyejin Shin, and Junbum Shin. Collecting and analyzing data from smart device users with local differential privacy. *arXiv preprint arXiv:1606.05053*, 2016.

[35] Victor Perrier, Hassan Jameel Asghar, and Dali Kaafar. Private continual release of real-valued data streams. *arXiv preprint arXiv:1811.03197*, 2018.

[36] Hossein Pishro-Nik. Introduction to probability, statistics, and random processes. 2016.

[37] Pasika Ranaweera, Anca Delia Jurcut, and Madhusanka Liyanage. Survey on multi-access edge computing security and privacy. *IEEE Communications Surveys & Tutorials*, 23(2):1078–1124, 2021.

[38] Muhammad Usman, Mian Ahmad Jan, and Deepak Puthal. Paal: A framework based on authentication, aggregation, and local differential privacy for internet of multimedia things. *IEEE Internet of Things Journal*, 7(4):2501–2508, 2020.

[39] Israel C Vidal, Franck Rousseau, and Javam C Machado. Achieving differential privacy in smart home scenarios. In *Anais do XXXIV Simpósio Brasileiro de Banco de Dados*, pages 211–216. SBC, 2019.

[40] Shupeng Wang, Jun Li, Guangjun Wu, Handi Chen, and Shihui Sun. Joint optimization of task offloading and resource allocation based on differential privacy in vehicular edge computing. *IEEE Transactions on Computational Social Systems*, 2021.

[41] Tian Wang, Yaxin Mei, Weijia Jia, Xi Zheng, Guojun Wang, and Mande Xie. Edge-based differential privacy computing for sensor–cloud systems. *Journal of Parallel and Distributed computing*, 136:75–85, 2020.

[42] Xingxing Xiong, Shubo Liu, Dan Li, Zhaohui Cai, and Xiaoguang Niu. A comprehensive survey on local differential privacy. *Security and Communication Networks*, 2020, 2020.

[43] Muktar Yahuza, Mohd Yamani Idna Bin Idris, Ainuddin Wahid Bin Abdul Wahab, Anthony TS Ho, Suleman Khan, Siti Nurmaya Binti Musa, and Azni Zarina Binti Taha. Systematic review on security and privacy requirements in edge computing: State of the art and future research opportunities. *IEEE Access*, 8:76541–76567, 2020.