DPWGAN: High-Quality Load Profiles Synthesis With Differential Privacy Guarantees

Jiaqi Huang , Qiushi Huang, Gaoyang Mou, and Chenye Wu

Abstract——

Smart meters collect more detailed information about how much electricity people use, helping to improve how the smart grid works. But there' s a big problem: this data can reveal private information about people, much to the public' s dismay. Current solutions attempt to protect privacy by slightly altering the data or creating fake data for analysis! But these methods are not always effective. So, we suggest you use a new, super cool technique called differentially private Wasserstein Generative Adversarial Networks (DPWGAN). This method privatizes real data and turns it into fake data; Researchers can therefore search anonymously. Our experiments show that this approach works really well for studying electricity consumption without compromising privacy.

Index Terms—Data synthesis, GAN, differential privacy, load profiling.

I. INTRODUCTION

So in lots of parts of south america. A loss of records that stops weight problems from being analyzed properly.privacy problems may lead different different problems so that person:s daily habits may get leaked.

Now, stepping into the nitty gritty, the global smart meter set up sport may want to reach approximately 200 million devices by 2030" s. That' s an entire bunch, proper? With those smart meters, we have been talking crazy memories about how human beings use energy and this facts is like gold for things like load forecasting, identifying consumerbehavior and demand management.. All of this exploratory human power can spill the beans on some privateness issues, like their lives and device ownership. And human beings are apprehensive about it, making them reluctant to own these clever meters and proportion their utilization statistics

Now, there are two main methods of doing those by removing elements and with the possibility of decreasing factors. First, you can see the data and cover easy such things as blurs or changes. This is generally known to as a data perturbation mechanism, we use it for deleting and adding some noise so that data privacy is safe and it' s all approximately the usage of privatedata analytics. However, there may be some issues this method frequently requires customisation for each particular thing.Our main plan is to construct something that analyzes private weight facts exceptionally effortlessly. Even when we blur something for example we have blurred phone numbers and kept the data some people may combine data with aadhar number and birthdate and find all the data. So it is

not possible in some cases and inefficient.Second, you can create false weight statistics that looks similar to original ones which cant be found by general people. You appear to be processing records in an smart way anyway. Public opinion became fresh with the usage of Generative Adversarial Networks (GAN), but it turned out that, notwithstanding their glamorous image, there has been nevertheless a danger of personal facts leaking out.

Therefore, we gift this other version, combining GAN abilities with rock-tough privacy that offers firstrate privateness. This way, we get first class of both worlds accurate responsibility and privacy at the door. Other jobs

There are two main ways to analyze weight data by keeping people' s data private.

. The aim of this paper is to create a methodology for analyzing privacy burden data in an alternative and not necessarily elaborate way.

The first type, known as data perturbation, is the use of pre-analytical or analytical data to hide important information. One popular approach is differential privacy, which ensures privacy by adding a controlled amount of noise to the data. but for almost any particular application existing methods must be modified

The second method which provides weight data that are as accurate as possible it is believed to lead to privacy, but recent research has shown that even complex technologies like Generative Adversarial Networks (GANs) can leak privacy

To address these challenges, we propose a new algorithm that provides imaginary GANs with strong privacy guarantees for discrete privacy. This approach has given rise to two related streams of research: one focusing on ensuring privacy in load data analysis and the other on load data generation using GANs

PROBLEM STATEMENT: DPWGAN (Differentially Private Wasserstein Generative Adversarial Networks) aims to preserve each user's privacy in the original load dataset by creating the best synthetic dataset This means that generated data looks very much like real data that the privacy of individual users will remain protected.

Impact of common errors in differential privacy analysis

Different levels of privacy have been learned to ensure weight research is greatly exaggerated with strong privacy protections, y'know. Initially, the effort was focused on ensuring that privacy guarantees were provided at the individual device level, e.g., reading, on smart meters. For example, Zhao et al. An efficient random battery-based weight-hiding algorithm is proposed to ensure different privacy, hum. Based on this, Zhang et al; *Improved analysis by loss of confidentiality, which can* lead to cost savings under pricing schemes. Furthermore, similar to Wang et al., we investigated how different parameters of differential privacy affect the performance of non-intrusive load monitoring *(NILM)* techniques, slightly.

So, you see, improving error-prone privacy can really make a big difference in weight analysis, right?

Recent efforts have focused on maintaining privacy when studying the use of individual electrodes. For example, some researchers have developed techniques to preserve users' personal information during power surveys or troubleshooting power

problems.But unlike these techniques are usually limited to specific tasks, we the method provides synthetic power consumption data that is confidential and useful for detailed analysis.

Many studies now use computer models to generate realistic electricity consumption data. For example, some models have simple details, while others note concrete electrical details. But most of this research focuses on providing data to the consumer, as we try to generate fake data for all households at once, while preserving their privacy

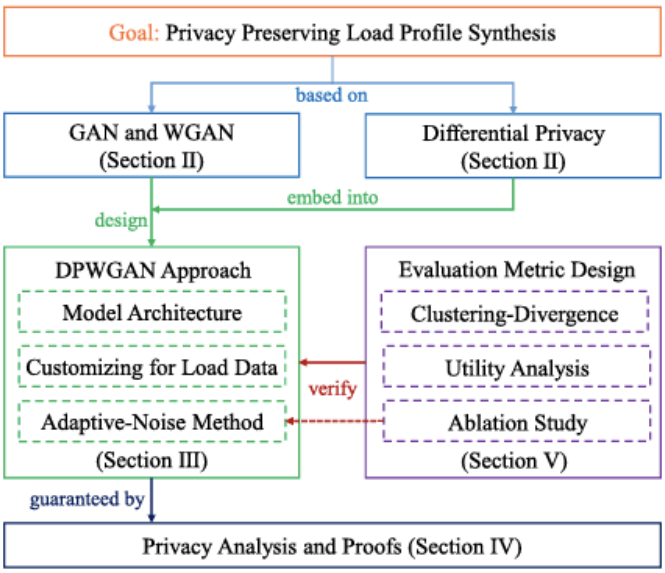
Our idea is to combine two different approaches: one is good at producing accurate data and the other is good at archiving. introduction

Our process ensures that the data we generate appears accurate but does not reveal personal information. We also found a way to measure the quality of our fake data against the real thing. The sections

* Some explain the basics of privacy and computer graphics
* Description Our approach is unique.
* We also suggest that our method guarantees privacy, which is important. Next, we investigate the efficiency of our method. Finally, we draw some conclusions.

In summary, when we look at how people use electricity in a country or country, to understand patterns, we need to ensure that their privacy is protected. Other methods cannot do this well, so we use a combination of other methods to store data p

.



Your Procedure or Your Method

Differential Privacy: We say to datasets are neighboring if one dataset is obtained by adding a record to other dataset. Differential privacy requires that any release of information about an input dataset should be done via a randomized algorithm M. such that the output of M does not reveal much information about any particular participant (tuple) in the input dataset. The output distributions of M should be similar on any two neighboring input datasets.

$$\Pr\left[\mathcal{M}(D_1) \in \mathcal{O}\right] \leqslant e^{\varepsilon} \cdot \Pr\left[\mathcal{M}(D_2) \in \mathcal{O}\right] + \delta.$$

In this paper, when we talk about a data set (denoted as D), we mean a group of individual samples, such as data from different people. We say that two data sets are "neighborly" if they are nearly identical, except that one observation may be present in one data set but not in the other.

What we call the probability (Pr) depends on how the method (M) works. Typically, we set a parameter named δ to 1 divided by the size of the data set (|D|).

For a given person, if we have some data sets affecting them and another not, then the two data sets are neighbors. This option ensures that whether a person's data is in the dataset or not, the results are the same. The privacy criterion in the equation tells us how likely this outcome is. Lower prices mean better privacy, but may need more noise to do so. This trade-off between privacy and performance is a great research topic.

The post-processing property of differential privacy states that no matter what we do with the output of our method, we should still maintain similar privacy

Lemma 1: Let $\mathcal{M}_1 : \mathcal{D} \to \mathcal{R}_1$ be a randomized algorithm that satisfies (ϵ, δ)-differential privacy. Let $f : \mathcal{R}_1 \to \mathcal{R}_2$ be an arbitrary randomized mapping. Then mechanism $\mathcal{M}_2 : \mathcal{D} \to \mathcal{R}_1 \times \mathcal{R}_2$ defined as $\mathcal{M}_2(D) = (\mathcal{M}_1(D), f \circ \mathcal{M}_1(D))$ also satisfies (ϵ, δ)-differential privacy.

For example, if we can generate synthetic load profiles with a differential privacy guarantee, then any analytical result derived from the synthetic data enjoys the same guarantee.

Definition 3 (Rényi Differential Privacy): A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (α, ϵ)-RDP if for every pair of neighboring datasets $D, D' \in \mathcal{D}$:

$$D_\alpha\big(\mathcal{M}(D) \| \mathcal{M}(D')\big) \leq \epsilon. \tag{2}$$

In DPWGAN, we inject Gaussian noise to achieve RDP. Specifically, we employ an RDP Sampled Gaussian Mechanism (SGM) bound from [23].

Lemma 2: Let $f : \mathcal{D} \to \mathcal{R}$ satisfies that $\|f(D) - f(D')\|_2 \leq \Delta$ for any neighboring datasets $D, D' \in \mathcal{D}$, then the Sampled Gaussian Mechanism $\mathrm{SG}(D) \triangleq f(\{x : x \in D \text{ is sampled w.p. } q\}) + \mathcal{N}(0, \sigma^2 \Delta^2 \mathbf{I}^d)$ satisfies $(\alpha, \epsilon(\alpha))$-RDP for any positive integer $\alpha > 1$ where $\epsilon(\alpha) = \frac{1}{\alpha-1} \log(\sum_{k=0}^{\alpha} \binom{\alpha}{k} (1-q)^{\alpha-k} q^k \exp(\frac{k^2-k}{2\sigma^2}))$.

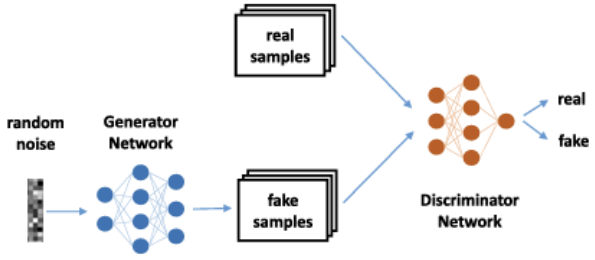
In addition, RDP can be converted back to (ϵ, δ)-differential privacy easily using the following lemma [22].

Lemma 3: If f is an (α, ϵ)-RDP mechanism, then it also satisfies $(\epsilon + \frac{\log 1/\delta}{\alpha-1}, \delta)$-differential privacy for any $0 < \delta < 1$.

The machine learning technique called Differentially Private Stochastic Gradient Descent (DPSGD) trains models while preserving the privacy of training data by adding small amounts of randomness to the model's updating process, thus no single datum can be left out.

Here we use random selection to sample from our data in order to train the model which means that estimates need a little tweaking so that no one point is affected too much. This procedure continues throughout the process of training.

Along this line, there is also need for us to think about privacy management. However, with Renyi Differential Privacy (RDP), we can achieve it in a simple way though it might appear complex



Privacy Loss (also called Privacy Leakage) refers to the extent to which privacy is surrendered when an individual's data is used in a randomizing process.

Assume we have two datasets, D and D', that are very close to each other (nearby datasets). We also have some more additional auxiliary information referred to as aux. Well then, let us assume that we run a randomizing process and obtain an outcome labelled as o.

At this outcome, o, the privacy loss is essentially a

measure of how much individualized information can be gotten from the dataset due to such an occurrence. It determines how much sensitive information about people can be exposed by randomization
process.

$$c(o; \mathcal{M}, aux, D, D') \triangleq \log \frac{\Pr[\mathcal{M}(aux, D) = o]}{\Pr[\mathcal{M}(aux, D') = o]}. \quad ($$

The Moments Accountant is a privacy-preserving mechanism tool used to analyze the privacy guarantees of randomized processes.

Here is what it includes:

Given a randomization mechanism M, neighboring datasets D and D′ with any additional auxiliary input aux, the Moments Accountant assesses particular statistical moments of an output distribution of this mechanism.

These mathematical moments help understand how individual privacy is impacted by this randomization mechanism. These moments offer ways to evaluate and quantify the level of safety provided by the mechanism under diverse data configurations and auxiliary information.

$$\alpha_{\mathcal{M}}(\lambda) \triangleq \max_{aux, D, D'} \alpha_{\mathcal{M}}(\lambda; aux, D, D'), \quad (4)$$

where $\alpha_{\mathcal{M}}(\lambda; aux, D, D') \triangleq \log \mathbb{E}[\exp(\lambda c(\mathcal{M}, aux, D, D'))]$ is the moment generating function of the privacy loss random variable.

Machine studying makes use of GANs (Generative Adversarial Networks) and WGANs (Wasserstein Generative Adversarial Networks), often to generate summaries that resemble actual data

Initially, GANs received attention due to their capacity to generate realistic models that don't exist in fact. Over the years, enhancements within the GAN algorithm have continuously advanced the great of the generated records, ensuing in loads of applications.

The major additives of a GAN are the generator network (GN) and the discrimination community (DN). GN is artificial statistics, whilst DN exams whether or not the information is proper or fake. This structure is shown in Figure 2 schooling method of GANs entails a dynamic interplay between GN and DN, much like a -player 0-sum game. The goal is to maximise the fee feature denoted as V(G,D) which represents the general overall performance of the GAN

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{\text{data}}(x)}\big[\log D(x)\big] + \mathbb{E}_{z \sim p_z(z)}\big[\log(1 - D(G(z)))\big].$$

. For The present code aims at estimating the Jensen Shannon (JS) divergence that assesses how close the two samples of real and generated data are. Or in other words, think of it as reducing variation between actual and artificial distributions from training.

The training process can be thought of as a game where the Generator Network (GN) generates realistic data while the Discriminator Network (DN) tries to distinguish between the fake ones and those that are real. The goal is to achieve Nash equilibrium whereby DN minimizes its loss whereas GN maximizes its loss.

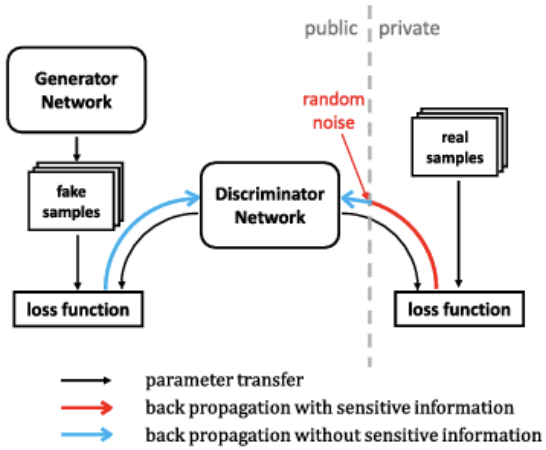
Conversely, JS divergence is prone to mode collapse (where generator produces very few types of samples) and it doesn’ t work well in practice hence increasing privacy costs. Therefore, instead of JS divergence WGANs were considered. These play games with each other based on an approximate Wasserstein distance which gives a measure of how much effort one needs to make for moving a

distributional towards another distributional.

This meant constraining DN' s parameters in WGANs so as to make them smooth. Here, this will help normalize learning thus having more statistical features about raw data by encouraging generator.

The training process is considered as a random mechanism, which takes a data set as input and outputs neural network parameters.

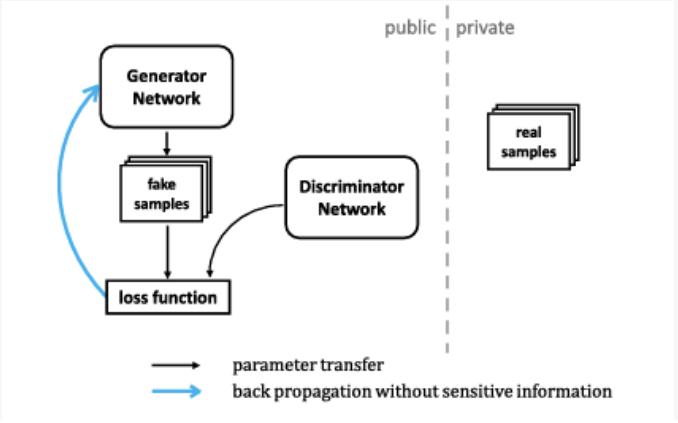
For confidentiality, the differential network (DN) is separately trained with a private stochastic gradient descent (DPSGD) algorithm. This introduces noise into the gradients calculated from real samples, preserves important information, and allows the network to detect useful features



Training follows a similar procedure for DPWGAN WGANs, but with additional steps for DPSGD. The WGAN loss function guides the training process by measuring the value of the discrimination network.

It should be noted that various private stochastic gradient descent (DPSGD) algorithms are widely used in training neural networks to protect individual data points, thus providing security measures to prevent unauthorized access spikes!! !

However, the differential privacy WGAN training process can be complicated due to the complexity of the differential networks .and additional steps required for safe training Despite all these challenges, THIS is a powerful tool for neurotraining for various functions.



In training a DPWGAN, the size of the networks matters. Small networks are more preferable because they converge faster. However, in terms of generating synthetic data that is very close to the original dataset, larger networks tend to be more useful. This is because larger networks can capture all the minute details in data. Thus, deciding on the network size really depends on how complicated the dataset is. For large and diverse datasets, larger networks are therefore required to fully grasp their intricacies. Nonetheless it should be noted that bigger nets come with a catch: extra privacy measures must be

put in place so as to maintain anonymity during training. This is because for larger gradients greater perturbations caused by noise have been observed while maintaining privacy during DPWGAN training activities. Therefore, striking balance between network size and privacy protection becomes critical for success of DPWGAN training.

The game changer in DPWGAN training is adaptive-noise method . traditionally noise scale has remained constant throughout like DPSGD; this however does not lead to optimal privacy preservation even if there are many iterations involved.The adaptive-noise technique; however make adjustments

On the other hand, dynamic adjusted noise addition in the course of training depending on how far along it is. If we start our training to make an optimization goal with fast progress adding noisy gradients would be enough. But as you go through the training and get closer to convergence, having noisy gradients could result into oscillations about the optima. In such cases, more accurate gradients are required for fine-tuning. Thus, it is logical to begin with a relatively large scale of noise in early phase of training so as to save privacy budget and then gradually reduce it for better convergence. This method has proven its efficiency in stable and effective DPWGAN training.

Now let us talk about privacy guarantees that DPWGAN offers. A key contribution of DPWGAN is each user in the original dataset having strong differential privacy guarantees individually. Differential privacy ensures that even when a model is trained using a dataset, individual data points remain safeguarded. Injected noise effectively hides information about any individual in DPWGAN making it theoretically impossible for models to learn specific details about individuals' data points due to lack of clarity around these issues. Nevertheless, noise impacts on collective information

To illustrate how DPWGAN in Algorithm 1 satisfies differential privacy, we examine each iteration process. This algorithm consists of a step by the Discriminator and a Generator step. In Discriminator step, synthetic samples {u_1,… .,u_n} are generated using the generator θ (t-1) from the previous iteration which has the same privacy guarantee as θ (t-1). The discriminator parameters are then updated using both these synthetic samples and real samples from dataset D that results into ω(t). We refer to the entire procedure of Discriminator step as M_DS(D,ω(t-1),θ (t-1)) = ω(t). That is followed by Generator step where we generate synthetic samples via θ (t-1) and calculate gradients for Generator parameters using ω(t). These gradients update the Generator parameters and result into θ (t). We denote this procedure as M_GS(ω(t), θ (t-1)) = θ (t). Thus, overall Algorithm 1 iteration can be expressed as:

M(t) = M_DS(D,ω(t-1),θ (t-1))

M_GS(M_DS(D,ω(t-1),θ (t-1)),θ (t-1)) = (ω(t),θ (t))

It should be noted that data set D is specifically inputted into .(M(t)\) as well as outputs from a previous iteration making it possible to apply lemma 4 during combining privacy used.

$$\begin{aligned}
&\mathcal{M}^{(t)}\left(D, \omega^{(t-1)}, \theta^{(t-1)}\right) \\
&= \left(\mathcal{M}_{DS}\left(D, \omega^{(t-1)}, \theta^{(t-1)}\right),\right. \\
&\qquad \left.\mathcal{M}_{GS}\left(\mathcal{M}_{DS}\left(D, \omega^{(t-1)}, \theta^{(t-1)}\right), \theta^{(t-1)}\right)\right) \\
&= \left(\omega^{(t)}, \mathcal{M}_{GS}\left(\omega^{(t)}, \theta^{(t-1)}\right)\right) \\
&= \left(\omega^{(t)}, \theta^{(t)}\right).
\end{aligned}$$

Lemma 5: Let $\mathcal{M}: \mathcal{D} \to \mathcal{R}$ be a randomized mechanism. For any $\lambda > 0$, suppose the moments accountant of $\mathcal{M}$ is $\alpha_{\mathcal{M}}(\lambda)$, then $\alpha_{\mathcal{M}}(\lambda) \geq \alpha$ is equivalent to $\mathcal{M}$ satisfying $(\lambda+1, \frac{\alpha}{\lambda})$-RDP for any $\alpha > 0$.

Proof: For any neighboring datasets D and D', we have

$$\begin{aligned} & D_{\lambda+1}\big(\mathcal{M}(D)||\mathcal{M}(D')\big) \\ &= \frac{1}{\lambda}\log \mathbb{E}_{o\sim\mathcal{M}(D')}\left(\frac{\Pr[\mathcal{M}(D)=o]}{\Pr[\mathcal{M}(D')=o]}\right)^{\lambda+1} \\ &= \frac{1}{\lambda}\log \int_{\mathcal{R}} \frac{(\Pr[\mathcal{M}(D)=x])^{\lambda+1}}{(\Pr[\mathcal{M}(D')=x])^{\lambda}}\,dx \\ &= \frac{1}{\lambda}\log \mathbb{E}_{o\sim\mathcal{M}(D)}\left(\frac{\Pr[\mathcal{M}(D)=o]}{\Pr[\mathcal{M}(D')=o]}\right)^{\lambda} \end{aligned}$$

Algorithm 1 DPWGAN

Require: Privacy target ϵ_0 and δ_0, load dataset D, weights bound of WGAN c_p, batch size n, noise scale σ, learning rate of Discriminator α_d, learning rate of Generator α_g, gradient norm bound C.

Ensure: Differentially private WGAN generator θ.

1: Initialize Discriminator parameters $w^{(0)}$ and Generator parameters $\theta^{(0)}$, set step number $t = 0$, privacy cost $\hat{\epsilon} = 0$, set moments accountant $\alpha(l) = 0$ for $l = 1, \ldots, L$.
2: **while** $\hat{\epsilon} < \epsilon_0$ **do**
3: $t \leftarrow t + 1$
4: **Discriminator Step:**
5: Generate synthetic samples $\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ using Generator $\theta^{(t-1)}$.
6: Sample $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \overset{\text{i.i.d.}}{\sim} D$ from the load dataset.
7: For each i, $\mathbf{g}_w(\mathbf{x}_i) \leftarrow \nabla_{w^{(t)}}(f_{w^{(t-1)}}(\mathbf{x}_i))$.
8: For each i, $\mathbf{g}_w(\mathbf{u}_i) \leftarrow \nabla_{w^{(t)}}(f_{w^{(t-1)}}(\mathbf{u}_i))$.
9: For each i, $\overline{\mathbf{g}}_w(\mathbf{x}_i) \leftarrow \mathbf{g}_w(\mathbf{x}_i)/\max\left(1, \frac{\|\mathbf{g}_w(\mathbf{x}_i)\|_2}{C}\right)$. ▷ Clip gradients.
10: $\mathbf{g}_w \leftarrow \frac{1}{n}\left(\sum_{i=1}^{n}(\overline{\mathbf{g}}_w(\mathbf{x}_i) - \mathbf{g}_w(\mathbf{u}_i)) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$. ▷ Noisy aggregation of gradients.
11: Update the moments accountant: $\alpha(l) = \alpha(l) + \log\left(\sum_{k=0}^{l+1}\binom{l+1}{k}(1-q)^{l+1-k}q^k \exp\left(\frac{k^2-k}{2\sigma^2}\right)\right)$ for $l = 1, ..., L$, where $q = \frac{n}{|D|}$ is the sampling rate.
12: $\hat{\epsilon} \leftarrow \min_l \frac{\alpha(l)-\log(\delta_0)}{l}$.
13: Update σ according to the adaptive-noise method.
14: $w^{(t)} \leftarrow w^{(t-1)} + \alpha_d \cdot RMSProp(w^{(t)}, \mathbf{g}_w)$.
15: $w^{(t)} \leftarrow \text{clip}\left(w^{(t)}, -c_p, c_p\right)$ ▷ Weights Clip.
16: **Generator Step:**
17: Generate synthetic samples $\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ using Generator $\theta^{(t-1)}$.
18: $\mathbf{g}_\theta \leftarrow -\nabla_{\theta^{(t-1)}}\frac{1}{n}\sum_{i=1}^{n} f_{w^{(t)}}(\mathbf{u}_i)$.
19: $\theta^{(t)} \leftarrow \theta^{(t-1)} + \alpha_g \cdot RMSProp(\theta^{(t-1)}, \mathbf{g}_\theta)$.
20: **end while**
21: **return** $\theta^{(t-1)}$.

In simple terms, Lemma 5 stipulates that the moments accountant in Renyi Differential Privacy (RDP) is fundamentally equivalent to a constant multiplier. Thus, it means that we can update the moments accountant for every iteration of our algorithm by summing privacy cost for all previous iterations with that of the current one thus facilitating easy composition of privacy guarantees.

To find the privacy consumption per iteration \(M(t) \), we divide the process into two steps: Discriminator step \(M_{DS} \) and Generator step \(M_{GS} \). In Disciriminator step \(M_{DS} \) we updated Discriminator parameters using sampled Gaussian Mechanism ensuring differential privacy. On the other hand, in Generator step \(M_{GS} \), Generator parameters are updated depending on output from \(M_{DS} \) together with previous parameters implying same level of privacy due to post-processing property.

For example, considering equation (9) which describes Gradient Noising Aggregation process:

delta= C/ n

Whereas, each gradient' s l2 norm is bounded by some C such that noisy aggregation behaves like a sample Gaussian mechanism with privacy parameter

$$\begin{aligned} &= \frac{1}{\lambda}\log \mathbb{E}_{o\sim\mathcal{M}(D)}\left[\exp\left(\lambda \cdot \log\frac{\Pr[\mathcal{M}(D)=o]}{\Pr[\mathcal{M}(D')=o]}\right)\right] \\ &= \frac{1}{\lambda}\log \mathbb{E}_{o\sim\mathcal{M}(D)}\left[\exp\left(\lambda c\big(\mathcal{M}, D, D'\big)\right)\right] \\ &= \frac{\alpha_{\mathcal{M}}\big(\lambda; D, D'\big)}{\lambda}. \end{aligned}$$

Therefore, we have

$$\max_{D,D'} D_{\lambda+1}\big(\mathcal{M}(D)||\mathcal{M}(D')\big) = \frac{1}{\lambda}\max_{D,D'}\alpha_{\mathcal{M}}\big(\lambda; D, D'\big),$$

h

Lemma 6: If $f : \mathcal{D} \to \mathcal{R}$ satisfies that $\|f(D) - f(D')\|_2 \leq \Delta$ for any neighboring datasets $D, D' \in \mathcal{D}$, then the moments accountant of Sampled Gaussian Mechanism $\mathrm{SG}(D) \triangleq f(\{x : x \in D \text{ is sampled w.p. } q\}) + \mathcal{N}(0, \sigma^2 \Delta^2 \mathbf{I}^d)$ satisfies that

$$\alpha(l) \leq \log\left(\sum_{k=0}^{l+1} \binom{l+1}{k} (1-q)^{l+1-k} q^k \exp\left(\frac{k^2 - k}{2\sigma^2}\right)\right) \quad (10)$$

for any integer $l \geq 1$.

i

RESULTS

Quality and usefulness of the dataset generated under various levels of differential privacy protection are assessed in this section to evaluate privacy-utility trade-offs. Furthermore, it is our objective to stress the real-world applicability of synthetic data set as an alternative to its original version. To begin with, we outline our experimental setup.

Implementation Details:

The Pecan Street Dataset will be referred to throughout this study since it contains 1-minute resolution data from 346 users between January 31st and April 30th, 2016. Data for each user was divided into daily intervals so that every sample is of equal length (1440). To create a situation typical for many users, we decided that one of each 1440-length sample represents an individual user' s record thereby creating tens of thousands such samples per day. Any point with a consumption rate above or equal to 10 was considered an outlier and therefore removed from the data. Preprocessing left slightly more than thirty thousand cases out of which we randomly chose thirty thousand for our experiments on private load dataset. For implementation purposes, Opacus library is used while the detailed network structures are illustrated in Figure 4 according to Section III-B guidelines.

Trade-Off Between Privacy and Utility:

Private preserving

Figure 5 shows that the levels of differential privacy protection provided may not have tight enough guarantees for real-world use as ε , a typical choice in most differential privacy studies, is less than 10. But exceeding the privacy parameter should not be seen as a flaw in what we are doing here. By reducing the dimension of training samples or increasing the number of users in the dataset, this parameter can be adjusted. It was decided to keep a high dimension of training samples so that DPWGAN could produce large and complex samples more effectively.

Clustering-Divergence for Synthetic Load Profile Evaluation

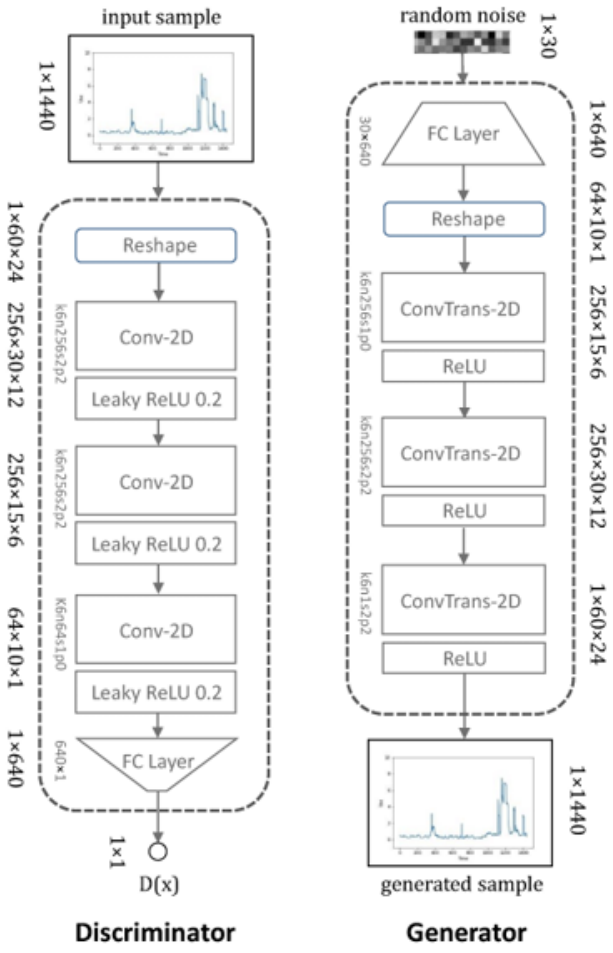
This is known as "privacy-utility trade-off" in privacy field research. In order to protect privacy completely, it is better not to use sensitive data at all. Whereas representing effects of the privacy preserved approach must take care about both effectiveness for privacy protection and utility associated with it. All experimental results throughout this section derive from comprehensive evaluations conducted to demonstrate how utilities corresponding to different levels of differential privacy vary.

DPWGAN synthetic samples under different levels of differential privacy are shown in Figure 5. These are the nearest neighbors for each sample in the original data with respect to Euclidean distance. Among other things, we reproduced a well-known effect that some samples produced by non-private GANs are similar to

certain training examples. On the contrary, synthetic datasets generated by DPWGAN largely wipe away individualistic patterns, thus implying its capacity to preserve privacy information of users. Irrespective of how much noise is injected into it, this model consistently stably generates synthetic data even at high noising levels.

The quality assessment of time-series data created by Generative Adversarial Networks (GANs) is challenging especially images that change with time. There exist performance measures for artificial images like Inception score but there is lack of research on evaluating synthetic time-series data. Therefore, it would be preferable to establish an appropriate and representative yardstick for assessing the quality of synthesized load profiles.

The goal of evaluating the synthetic load dataset is ultimately whether common power consumption patterns among users are retained. However, differentiating between common and unique personal patterns tends to be difficult and the required patterns differ depending on what one is trying to achieve hence it becomes difficult to follow through. When evaluating this matter, we return to the origin of these patterns: the statistical distribution of consumption. This is a continuous random variable for power consumption distribution that is not directly represented in finite data samples hence discretization of random variable using clustering results and feature extraction. Non-obvious insights such as behavioral and technological drivers can be revealed by load clustering as a temporal assessment tool for residents' electricity usage. Therefore, K-means clustering adopts Euclidean distance to measure similarity between load series to group similar patterns together. There might be some similarities among users within the same cluster in terms of their lifestyle, dwelling places, and family background. In this case, clustering centers represent users while numbers of users in each cluster indicate dataset' s data distribution. The probability distribution frequency from samples falling into each cluster serves as robust approximation for comparing different datasets with respect to the given random variable' s probability distribution



abbrevia

The quality of a dataset is highly determined by how much its distribution closely resembles the original dataset which is known as Clustering-Divergence. For us to estimate this metric, we need to carry out K-means clustering on the original dataset and allocate users into different

clusters, then save the number of users in each cluster that is represented as distribution of initial data set P_o = {f_1, f_2, ..., f_k} with k being the number of clusters and fi being frequency of users belonging to cluster i. Also, for synthetic data sets distributions are obtained in terms of where about the clustering centers for the preexisting data set are situated that is given by Ps={f1,f2,… .,fk}. The Kullback-Leibler (KL) Divergence formula can be used for calculating Clustering-Divergence between Ps and Po:

It should be clearly established that KL-Divergence computation is not symmetrical thus Po comes after Ps because there may exist zero elements in Ps while non zero frequencies exist in Po such that divergence KL-divergence will be undefined.

We first verify our claim using a non-private GAN during experiments. During training process model learns

$$P_s = \{f_1', f_2', \ldots, f_k'\}. \tag{14}$$

We can employ the KL-Divergence to calculate the Clustering-Divergence between P_s and P_o

$$KL(P_s||P_o) = \sum_{i=1}^{k} f_i \log \frac{f_i}{f_i'}. \tag{15}$$

e non-private GAN. This wider range denotes a more varied synthetic dataset in DPWGAN than in non-private GAN. Though it does not seem to make sense in relation to privacy concerns, it is a deliberate compromise leading to differential privacy. Individual patterns are obscured by introducing more variability thereby increasing the protection of privacy.

Moreover, the choice of network structures and hyper-parameters plays a crucial role in the performance of DPWGAN. We kept the same network structures and hyper-parameters for both the non-private GAN and DPWGAN in our experiments so as to facilitate a faircomparison. Nevertheless, DPWGAN' s slightly slower convergence rate is good enough. The injected noise which interferes with gradients during training has been linked to this slower convergence speed. However, this effect is mitigated through use of large batch size in DPWGAN since aggregated gradients still maintain significant magnitudes that ensure that injected noise remains small enough. In such instances, this ensures that correct signs for gradients are preserved thus preventing very slow convergences by DPWGAN

To summarize, Clustering-Divergence metric serves as an important tool for assessing how well do we preserve the distributional characteristics of the initial dataset under consideration together with its necessary variability

In this section, we demonstrate empirically the importance of the adaptive-noise method through an ablation study, DPWGANs' performance being compared with and without adaptive noise scales. The aim is to illustrate how adaptive noise strategy enhances stability and privacy consumption during training.

For models which do not possess adaptive noise scales, we check the values of noise scale \(\sigma \) in 0.4, 0.5, 0.6 and 0.7 by using a model. On the other hand, when there are adaptive noise scales in the models, we make initialization at 1.5 and then apply exponential decay during training cycles too. Particularly, for every step when it is greater than 0.4, decreased by 2% while for any step between 0.3 and 0.4 reduced by 1%. When \(\sigma \) attains a value of less than or equal to zero point three (≤0.3), its decay ceases.

We perform training ten times and compute average performances employing Clustering-Divergence as a measure thereof.The results indicated on Figure 7 show that under an Adaptive-Noise plan our model exhibits better stability and consumes low privacy during convergence process compared to fixed noise models.Though there is a considerable increase in performance for the first few epochs with constant noise scale values at around this time the model fails to converge leading to poorer performances overall since more privacy was needed for learning later on; all these effects were evident especially when \(\sigma =\)0.5 – \(\sigma =\)0.6 but even still true when σ =0 .7(Figure nulber). For σ =

On the contrary, we initialize the noise scale at 1.5 and apply an exponential decay during training to models with adaptive noise scales. That is, if it is over .4 the noise scale will be decayed by 2% after each step while if it is between 0.3 and 0.4 it deteriorates by 1%. This continues until the noise scale hits a value of .3.

We train ten times and measure average performance using Clustering-Divergence. As shown in Figure 7 below, when model stability increases and privacy consumption decreases with respect to fixed noise models particularly under adaptive-noise strategy. In addition, fixed noise scale would initially improve the model' s performance; however, it would fail to converge later leading to worsened performance as well as increased privacy consumption as training process goes on (Fig5.a). We can observe this for

σ =0.5, σ =0.6 and σ =0.7 equalities of standard deviation in figure5.b Even for

σ =0.4 there is still a possibility for improvement of the model because its performance isn' t at its peak yet that shows imperfect convergence (Fig5(b)). This is a limitation that requires careful selection of the initial noise scaling before training takes place in order for it not to hinder researchers significantly (Fig6(a). It should be noted that such situations do not arise when no pulse is used while creating a model indicating that reducing noise can also enhance quality graphically.Furthermore this naturally brings us to consider tuning our no

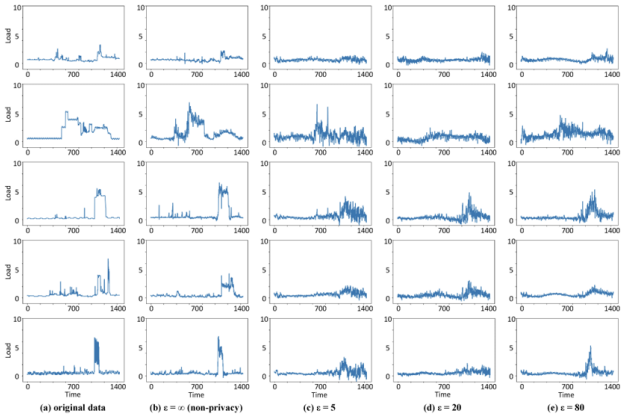


Fig. 5. Synthetic load data with different privacy levels using DPWGAN. Samples selected from the synthetic datasets are the nearest neighbors (w.r.t. l_2-norm) of the corresponding leftmost original samples.

This section is aimed at verifying whether the dataset of synthetic data generated by DPWGAN can be substituted for real world application, particularly in load profiling. We perform a simple load profiling exercise on both the original and synthetic datasets to evaluate the similarity between their outcomes. The close approximation of these results implies that the synthetic dataset may be used instead of the original one for this task thus confirming that DPWGAN is effective.

Consequently, it is reasonable to compare results

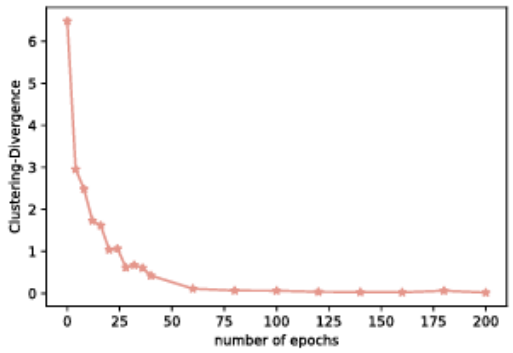
from different datasets. However, comparing results across different datasets is logical because it considers cases such as clustering high-dimensional data directly (for example, applying K-means clustering on load data) which could be substantially affected by initial point selection.

Consider a case where researchers want to know how total energy consumption by users varies during a day. One way to estimate this distribution is by counting how many samples fall into each bin, dividing the range of values of users' energy consumption into equal-length intervals, say 10. Similarity between the outputs produced from various kinds of datasets can then be measured through total variation distance;

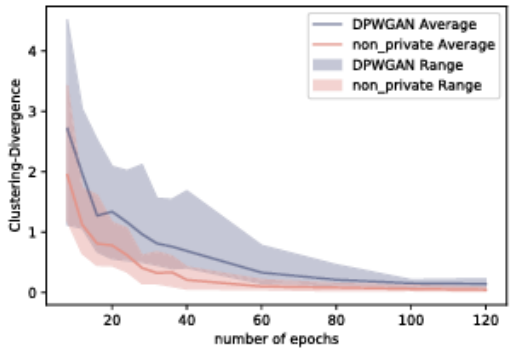
The similarity between the results acquired from one-of-a-kind datasets can then be assessed the use of the entire version distance, one of the maximum generally used statistical distances. A smaller distance implies a more similarity between the distributions, and vice versa. This metric allows us to gauge how intently the artificial dataset mirrors the authentic dataset's traits in phrases of power intake distribution

Our experiments comprised of fifteen attempts at each privacy level repeating the DPWGAN approach, where we calculated total variation distance for every attempt as illustrated in Figure 8. Furthermore, we presented a non-private GAN which acts as a standard measure. This is the WGAN without any privacy preservation measures but has similar network architectures with DPWGAN. Also, this graph also represented an ideal performance of the task indicating its lowest possible value that does not depend on privacy levels. As such, these curves remain horizontal in Fig 3 due to their non-private and ideal performances respectively.

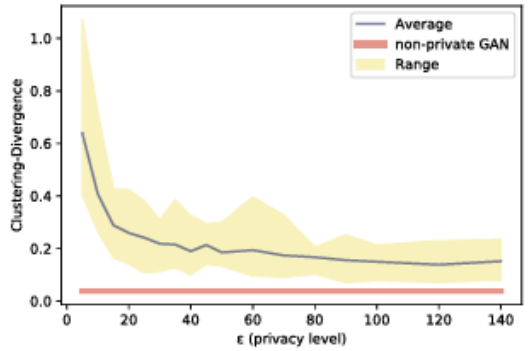
Sometimes our findings showed that right from the start of training, model performed good while sometimes it took sometime before converging. In general, DPGAN performed almost similarly to non-private GAN at right range of privacy levels and it approached to an ideal performance as we expected. Therefore, datasets generated by DPWGAN for load profile generation task showed potential to yield similar results with those obtained from the original dataset



(a) Clustering-Divergence tracking of a non-private GAN during training.



(b) Converge rate comparison between a non-private GAN and a DPWGAN during training.



(c) Clustering-Divergences of DPWGAN under different privacy levels.

Fig. 6. Using Clustering-Divergence to evaluate the synthetic dataset's quality. Note that the horizontal axis label of the first two figures is different from the third one's.

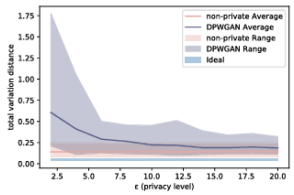


Fig. 8. The performance of DPWGAN on a load profiling task compared with a non-private GAN as well as the ideal performance, measured by the total variation distance.

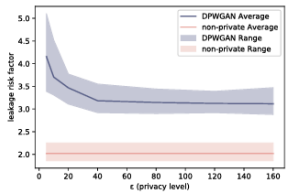


Fig. 9. The leakage risk factors of DPWGAN under different privacy levels.

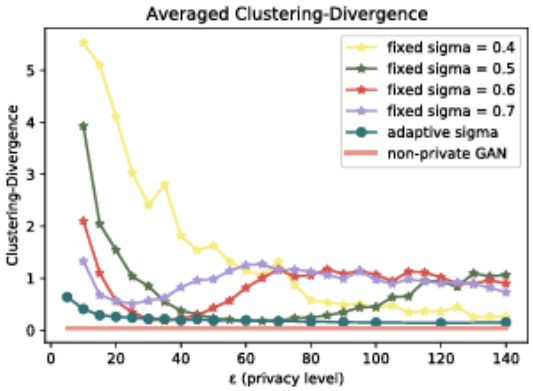


Fig. 7. The average performances of models with different noise scheduling strategies are presented. Datasets generated by adaptive-noise DPWGAN have higher and more stable performance than the fixed-noise models'.

Conclusion:

In this paper, the DPWGAN approach is introduced as a solution to generate load profiles with differential privacy guarantees. This means that if any analysis is performed on the synthetic data it will satisfy the differential privacy criterion and can be used even by non-experts in differential privacy. In order to evaluate the quality of the generated dataset, we suggest a new indicator namely Cluster-Divergence designed specifically for load data. Similarly, experimentation reveals why adaptive noise scale tuning is important to achieving optimal performance of the DPWGAN approach. As such, the results confirm that DPWGANs produce high-quality and practical data.

Acknowledgement: The preferred spelling of the word "acknowledgment" in American English is without an "e" after the "g." Use the singular heading even if you have many acknowledgments. Avoid expressions such as "One of us (S.B.A.) would like to thank ... ." Instead, write "F. A. Author thanks ... ." Sponsor and financial support acknowledgments are placed in the unnumbered footnote on the first page.And thanks to my professor Pradyumn K pandey for giving me this opportunity to solve this research paper…