



Privacy-Preserving Synthetic Smart Meters Data

Ganesh Del Grosso, Georg Pichler, Pablo Piantanida

► To cite this version:

Ganesh Del Grosso, Georg Pichler, Pablo Piantanida. Privacy-Preserving Synthetic Smart Meters Data. 2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Feb 2021, Washington, France. pp.1-5, 10.1109/isgt49243.2021.9372157 . hal-04137028

HAL Id: hal-04137028

<https://hal.science/hal-04137028>

Submitted on 22 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Privacy-Preserving Synthetic Smart Meters Data

Ganesh Del Grosso
INRIA, Ecole Polytechnique
Gif-sur-Yvette, France
ganesh.del-grosso-guzman@inria.fr

Georg Pichler
Technische Universität Wien
Vienna, Austria
georg.pichler@ieee.org

Pablo Piantanida
CentraleSupélec, CNRS, Université Paris-Saclay
Gif-sur-Yvette, France
pablo.piantanida@centralesupelec.fr

Abstract—Power consumption data is very useful as it allows to optimize power grids, detect anomalies and prevent failures, on top of being useful for diverse research purposes. However, the use of power consumption data raises significant privacy concerns, as this data usually belongs to clients of a power company. As a solution, we propose a method to generate synthetic power consumption samples that faithfully imitate the originals, but are detached from the clients and their identities. Our method is based on Generative Adversarial Networks (GANs). Our contribution is twofold. First, we focus on the quality of the generated data, which is not a trivial task as no standard evaluation methods are available. Then, we study the privacy guarantees provided to members of the training set of our neural network. As a minimum requirement for privacy, we demand our neural network to be robust to membership inference attacks, as these provide a gateway for further attacks in addition to presenting a privacy threat on their own. We find that there is a compromise to be made between the privacy and the performance provided by the algorithm.

Index Terms—Privacy, Smart Grids, Generative Adversarial Networks, Deep Learning.

I. INTRODUCTION

Power consumption data from individual households is essential for diverse research and operational purposes. It is already abundant and with the help of cheap smart metering devices very fine-grained analysis becomes possible. As machine learning (ML) algorithms can be implemented with ease, even on consumer hardware, and ML competitions flourish, energy providers might be tempted to take advantage of this opportunity. If an anonymized portion of their collected power consumption data is released publicly, scientists around the world can utilize it and they might uncover patterns or mechanisms that aid system operations. However, energy consumption data cannot be published or released, even when anonymized, without potentially violating the privacy of individuals. An attacker with enough side information to identify a particular household might be able to discern information, such as geographic location, active hours and even the specific appliances [6] present.

Our aim is to develop the means for generating synthetic power consumption data that faithfully imitates the important statistical properties of original data as an ensemble, even including variations and outliers. Then, this synthetic data can be used instead of real data for the planning of system operations and, as it is not tied to specific individuals, it can be distributed without privacy concerns. To approach this problem, we utilize Generative Adversarial Networks (GANs) [9], where, based on real power consumption data, an artificial neural network is trained to generate synthetic curves. This makes our task twofold: On the one hand, we need to carefully evaluate the generated data for its utility and on the other hand, we also need to assure that the generated curves do not enable an attacker to draw

conclusions about the individuals present in the original training set.

To address the issue of performance evaluation, two different methods are proposed. The first evaluation method relies on comparing the relevant statistical properties of artificially generated data to those of natural data. The second revolves around using artificial data for a prediction task

and comparing the results to those obtained by using natural data instead.

A Long Short-Term Memory (LSTM) network is trained for this purpose. This way of addressing the quality of generated data is common practice with GANs in the context of images, where artificial images are used to train a classifier network and the performance of this classifier is compared to a baseline trained with natural images [3].

Regarding privacy, membership inference attacks directly measure the privacy leakage of a machine learning model [16]. In view of this, we utilize three mechanisms to perform membership inference attacks, that are model agnostic within the GAN framework.

Different setups are considered, where we study the trade-off between privacy and performance of our algorithm by tuning the training process of the GAN.

A. Related Work

Generation of synthetic data. Research in generative models has seen immense progress in the past decade. In particular, there has been a lot of interest in the context of image processing for a myriad of tasks [10]. In this context, GANs are one of the most popular frameworks, as they have been shown to produce more natural-looking images than the alternatives. Previous works, [4], [5], proposed GANs for the generation of power curves; however, issues of privacy were not addressed. Sequential models, such as RNNs [17] and LSTMs [12], are often used for forecasting; yet, our task is to generate new instances of data that imitate the properties of natural data, without being tied to a particular instance of the original samples. Therefore, we focus purely on GANs, as they implicitly capture the underlying distribution of natural data, and allow us to sample new instances.

Membership inference attacks. ML algorithms can store features, or even entire records, that were used for its training [7], [8]. This means an attacker with access to an ML service could recover sensitive information about its training records. We want to prevent this and offer clean power consumption data for research or operational purposes. A minimum requirement for privacy is robustness to membership inference attacks. These attacks can identify members of the training set of an ML algorithm and use side information provided by the algorithm to determine

properties of the identified client. Thorough analysis of these attacks is available in [16], [18], [19]. However, little research has been done on the privacy leakage of generative models. A first example of a membership inference attack against a GAN was presented in [11]. Following that line of research, we propose several membership inference attacks to assess the privacy guarantees of our model.

B. Our contribution

GANs for sequential data generation: The model proposed in this paper is specially adapted for sequential data. It utilizes 1-dimensional convolutional layers to capture the data structure.

Evaluation using a forecasting task: Assessing the quality of synthetic data is a highly non-trivial task. In the case of images, popular methods involve training well established classification models using synthetic data. These models are known to perform well when trained with natural data; thus, the idea is to use the performance drop when trained with synthetic data as a quality measure of the generated data [3]. In the case of power consumption curves, there are no well established classification models. Furthermore, in our setup, data is not naturally divided into easily distinguishable classes. Consequently, we contribute a quantitative method for assessing the quality of synthetic data by training an LSTM network for forecasting. The principle is the same as for images; we compare the performance of the same LSTM, when trained with natural and synthetic data.

Evaluation using statistical properties of generated samples: As the generated samples should accurately represent the households in the training set, we select statistical features of the power consumption curves, which are commonly utilized to infer information about the households [2], [13]. In particular, artificial curves should closely resemble natural data in terms of their mean, coefficient of variation, skewness, kurtosis and maximum-mean ratio. The ensemble statistics of these “indicators” on the generated data should be close to the actual statistics. As a quality metric, we use the earth mover’s distance between the empirical distributions of the indicators on artificial and natural curves.

White-box Membership Inference Attacks: In [16], [18], an attack using an inference model is proposed which requires a dedicated training set. However, this may not be realistic and even unnecessary in some scenarios. We propose simpler attacks that exploit similar structural properties of the network, but without the need of a dedicated training set for the attacker. Our first attack directly utilizes the output of the discriminator, as proposed in [11], while the second attack utilizes the gradient of the loss function of the discriminator with respect to its model parameters, inspired by [16]. Both attacks compute a score for every input and select the top scoring input as the most likely candidate.

Black-box Membership Inference Attacks: All black-box attacks we have seen in the literature, including those proposed in [11], [16], [18], [19], are based on training a surrogate model that can be used to perform a white-box version of the attack. In general, this yields degraded results in comparison to the white-box attack. We propose to exploit the similarity between artificial and natural data to perform a black-box attack. Our black-box attack is inspired by our evaluation method using statistical properties and uses the

earth mover’s distance between the empirical distributions of artificial and natural curves in terms of their indicators.

Defense Mechanisms: We study the interplay between privacy and generalization, and how the choice of training hyper-parameters can protect against membership inference attacks. Additionally, we propose Gradient Norm Regularization, a defense mechanism designed to protect against membership attacks that exploit the gradient of the loss function.

II. PRIVACY VERSUS UTILITY TRADE-OFFS

A. Membership Inference

Membership inference consists of determining whether a particular example or set of examples were part of the training set of the target model. If an attacker cannot even determine membership, it is certainly not possible to obtain more detailed information. Thus, membership inference represents the weakest version of a violation of privacy. In our setup, we further simplify this task and provide 5 sets of examples to the attacker, asking for a prediction of which set was part of the training set. The attacker knows that one and only one of these subsets was part of the training set. To determine the accuracy of the membership inference attacks, we repeat the attack and report the ratio of successful attempts. A more detailed explanation is given in Section III. All the attacks presented in our work rely on computing a score for each subset, selecting the top scoring set as the most likely candidate. The rest of this section is dedicated to explaining the different scoring criteria.

1) *Likelihood Attack:* Recall that in the GAN setup, the job of the discriminator is to distinguish natural and synthetic samples. Intuitively, we predict that the discriminator will recognize samples, that were part of the training set, with higher confidence. The likelihood attack consists of querying the discriminator and using its average output (in $[0, 1]$, where 0: synthetic, 1: natural) to assign a score to each set of curves. The likelihood attack is detailed in

Algorithm 1 Likelihood Attack

Input: Universe \mathcal{U} , its partition $P_{\mathcal{U}}$, discriminator D .
Output: Index of the member of the training set i .
initialize likelihoodList $\leftarrow []$
for each set **in** $P_{\mathcal{U}}$ **do**
 $\bar{p} \leftarrow \text{mean}([D(\text{curve}) \text{ for each curve in set}])$
 likelihoodList.append(\bar{p})
end for
return argmax(likelihoodList)

Algorithm 1 where the average likelihood is denoted by \bar{p} . The attacker expects the discriminator to be more confident, i.e. output a higher likelihood, for curves that were seen during training. Thus, the attacker chooses the subset with the highest average likelihood.

2) *Gradient Norm Attack:* Neural Networks (NNs) utilize iterative optimization algorithms, such as Stochastic Gradient Descent (SGD), to minimize a loss function that serves a training goal. SGD computes the gradient of the loss function with respect to the NN’s parameters in order to minimize it. Since this is performed with training samples, the norm of the gradient of the loss function should be smaller at samples that were used during training, than on unseen samples. The gradient norm attack consists of

computing the norm of the gradient of the loss function with respect to the model parameters and assigning a score to each set of natural curves according to their average gradient norm, then the attacker selects the subset with lowest average norm. This is a white-box attack, in which the attacker must have access to the internal parameters of the target model. Algorithm 2 details the attack.

Algorithm 2 Gradient-Norm Attack

Input: Universe \mathcal{U} with partition $P_{\mathcal{U}}$, generator G , discriminator D_{θ} with discriminator parameters θ , Loss L .

Output: Index of the member of the training set i .

```

initialize normList  $\leftarrow []$ 
for each set in  $P_{\mathcal{U}}$  do
  initialize auxList  $\leftarrow []$ 
  for each curve in set do
     $g \leftarrow \nabla_{\theta} L(D_{\theta}, G, \text{curve})$ 
    auxList.append( $\|g\|_2$ )
  end for
  normList.append(mean(auxList))
end for
return argmin(normList)

```

3) *Indicators Attack*: This black-box attack exploits the similarity between generated and natural data in terms of the indicators. The idea behind this attack is that members of the training set will be closer to the synthetic samples in terms of these indicators. For each subset of natural curves, the attacker fetches the same amount of artificial curves, computes the indicators on both sets and compares them via the Earth Mover's Distance (EMD) of their empirical distributions. The set of natural curves with the smallest EMD is chosen. Let I be the vector-valued function that computes all indicators for a set of curves. Algorithm 3 shows the algorithm for the indicators attack.

Algorithm 3 Indicators Attack

Input: Universe \mathcal{U} , partition of universe $P_{\mathcal{U}}$, generator G , indicator functions I .

Output: Index of the member of the training set i .

```

initialize distList  $\leftarrow []$ 
for each set in  $P_{\mathcal{U}}$  do
   $z \leftarrow \text{randN}(0, 1)$  {Random input for generator.}
  setfake  $\leftarrow G(z)$ 
  AID  $\leftarrow \text{mean}(\text{EMD}(I(\text{set}), I(\text{set}_{\text{fake}})))$ 
  distList.append(AID)
end for
return argmin(distList)

```

B. Defense Mechanisms

Membership inference attacks rely on overfitting. When this happens the target model memorizes and reproduces samples in its training set, instead of learning their properties and underlying distribution. The likelihood attack exploits bias in the discriminator output, while the gradient-norm attack exploits the local behavior of the loss function around the training samples. On the other hand, the indicators attack relies on the overly precise imitation of statistical properties of the training set, rather than proper generalization to the whole universe. This reasoning indicates that generalization

could make the model more robust against the attacks presented here. To achieve good generalization, training hyper-parameters, such as the learning rates (LRs) for generator and discriminator, must be chosen carefully. Furthermore, the attacks could be incorporated in the training process and used to tune these hyper-parameters.

1) *Gradient Norm Regularization*: We propose a straightforward way to combat the gradient-norm attack, by adding a term to the standard loss that penalizes a small norm of the gradient with respect to model parameters. More precisely, we suggest the regularized loss:

$$L_{\text{reg}}(\theta) = L_{\text{discr}}(\theta) - \eta \|\nabla_{\theta} L_{\text{discr}}(\theta)\|_2, \quad (1)$$

where L_{discr} denotes the original discriminator loss, θ the parameters of the discriminator, and $\eta > 0$ is a Lagrangian.

III. NUMERICAL EXPERIMENTS

A. Data Set

Our experiments are based on ‘‘SmartMeter Energy Consumption Data in London Households’’ [1]. This set contains energy consumption readings for a sample of 5 567 London households that took part in the UK Power Networks led Low Carbon London project from November 2011 to February 2014. Power consumption readings were taken at half hour intervals in kWh (kilo Watt hours). For each reading a unique house identifier, date and time data and a CACI Acorn group label are provided. For our experiments we considered a mixed set of households with standard tariff from all CACI Acorn groups. We extracted two-week temporal frames, i.e., 672 consecutive power consumption readings, which are separated according to each unique household identifier.

B. Generative Model

To generate synthetic power consumption curves we propose a 1-dimensional convolutional GAN. The generator network has a latent space of length 42. The latent space input is sampled from a normal distribution. This choice for the latent space dimension leaves enough room to capture the variation present in the training set. The generator network is composed of a combination of linear and 1D up-convolutional layers and uses ReLU activation functions, save for the last layer, which uses tanh, to accord with the pre-processing of the data. The discriminator takes curves of length 672 as inputs. It is composed of a combination of 1D convolutional and linear layers, followed by ReLU activation functions, save for the last one, where a sigmoid is used instead. To avoid the vanishing gradient problem and make training more stable, spectral normalization is applied to all the layers of the discriminator, as proposed in [15]. The total complexity in terms of parameters is 1163909 for the the generator and 1143769 for the discriminator.

C. Evaluation Model

Synthetic data should provide the same utility as natural data when used for training an ML model for a certain task. We evaluate the quality of generated curves by training an LSTM model for forecasting using synthetic curves and testing its performance on a test set of natural curves. We train the same LSTM model using a portion of the training set of the generator and use this version of the model as a baseline for comparison. The LSTM architecture that we

chose takes 24 points of the curve at a time as input and predicts the subsequent 24 points (12 hours). The output of the LSTM is post-processed with a linear layer and a tanh activation function. The total complexity of this model is 15384 parameters. The performance of the LSTM is measured by Mean Squared Error (MSE) between prediction and ground truth on the test set. We compute the difference between the MSEs for the same LSTM, trained with artificial and natural curves. We refer to this difference as the LSTM score. The LSTM score can be negative when the LSTM trained with artificial curves performs better than the LSTM trained with natural curves.

For each curve we calculate the following indicators: mean, coefficient of variation, maximum-mean ratio, skewness, and kurtosis. The empirical distribution of each indicator over artificial and natural curves is computed. In our experiments we took the whole training set (around 2300 examples) and the same amount of generated curves for this purpose. The Earth Mover’s Distance (EMD) between the two histograms is used as a quality metric for evaluating the model. The average distance over all the different indicators is used to obtain a single quality index, which we call Average Indicator Distance. When computing the average distance over indicators, the indicators need to be comparable and hence require normalization. We normalize the indicators to unit variance, where the variance is estimated using both generated and natural data in order to prevent the normalization from distorting the result.

D. Experimental procedure

To study the interplay between performance and privacy leakage, we train our generative model in three different scenarios: i) the generator has a significantly higher learning rate (LR); ii) generator and discriminator share the same LR; iii) gradient-norm regularization is applied during training and generator and discriminator share the same LR. The attacker, however, is allowed to exploit the full loss of the discriminator, including the regularization term. We observe the impact of the LR on the quality and privacy of the model.

For each run of our experiments, we completely shuffle the universe of natural curves. We obtain a partition by assigning each household to one of five subsets. Each subset contains the same number of households and a comparable amount of curves (around 2300 each; the number of samples varies slightly from household to household). One subset is used for training and to assess the quality of the model in terms of the indicators. The other four subsets will play the role of non-members when performing the attacks. One of these four subsets is also used as test set to evaluate the generator using the LSTM approach.

The generative model is trained using the GAN algorithm [9]. Each run the model is trained for 140 epochs with Adam [14], $LR = 10^{-4}$ for both generator and discriminator in the same LR (labeled “same LR”) case and batch size 20. In the case of different LR (labeled “diff LR”), the LR of the generator is 10^{-4} and the LR of the discriminator is 10^{-5} . For the regularized case, η was set to 10^{-2} .

To compute the LSTM score, we train two identical LSTMs: on training data and on the generated curves. Both LSTMs are trained for 40 epochs using Adam [14], with $LR = 10^{-4}$ and batch size 50. The loss function for training is a combination of the MSE and the l_1 norm of the

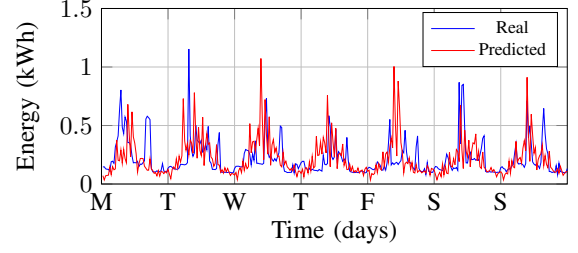


Fig. 1: Prediction by a model trained with artificial curves.

difference between statistics of the prediction and the ground truth. The statistics considered for the loss are the mean and the second, third and fourth centralized moments of the curves. This combination is optimized in order to obtain the best performing LSTM. After training, both LSTMs are evaluated on the test set using the average Mean-Squared-Error. Then, the difference between the average MSEs of the fake LSTM (trained with artificial curves) and real LSTM (trained with natural curves) is the LSTM score. To compute the Average Indicator Distance, we evaluate the indicators (mean, coefficient of variation, maximum-mean ratio, skewness and kurtosis) for all curves in the training set, and for the same number of generated curves. The Earth Mover’s distance is evaluated between distributions of indicators of natural and artificial curves, then normalized and averaged to obtain the Average Indicator Distance.

The attacker is given 5 subsets and performs the attack on subsets as a whole, subsequently labeled “per-subset”. The objective of the attacker is to determine which of the 5 subsets was the training set. In the “per-household” attack, a single household from each subset is selected randomly. The attacker is then given the data of these five households, one from each subset, and asked to determine which household was part of the training set. The per-subset attack are performed once per run, but per-household attacks are performed multiple times, by randomly picking different households.

E. Results

Table I shows the success rate for different attacks evaluated over ~ 100 runs. Attacks were performed once per run in the per-subset case and 100 times per run in the per-household case, randomly selecting different households each time. The indicators attack was only performed on a per-subset basis. LSTM score and Average Indicator Distance are reported for each subset with mean values over runs, and the error is the standard deviation. Having five subsets to choose from, a random guess would achieve 20% accuracy. Training the model with the same learning rate (ii) achieves the best performance in terms of the LSTM score and Average Indicator Distance; however, it also results in the most vulnerable model in terms of privacy. Modifying the LR (i) results in significant gains in privacy, at the cost of marginal losses in performance.

As expected, per-household attacks are overall weaker than per-subset attacks. This means that these attacks depend on the ensemble statistics of large groups of samples in order to be effective. Gradient-norm regularization (iii) provides additional robustness against the gradient-norm attack, but the sacrifice in performance is high compared to the slim gain that could be obtained by simply tuning the LR, as in (i). Likelihood attacks are more robust against defense

Scenario	i) diff LR	ii) same LR	iii) Regularized (same LR)
Per-subset gradient-norm attack	28.50%	68.50%	26.50%
Per-household gradient-norm attack	21.77%	27.27%	20.68%
Per-subset likelihood attack	39.00%	62.50%	39.50%
Per-household likelihood attack	23.01%	27.33%	22.61%
Indicators attack	24.50%	33.00%	24.50%
LSTM score	0.011 ± 0.013	0.002 ± 0.007	0.016 ± 0.012
Average Indicator Distance	0.39 ± 0.14	0.29 ± 0.09	0.50 ± 0.18

TABLE I: Accuracy of membership attacks, LSTM score and Average Indicator Distance for different setups.

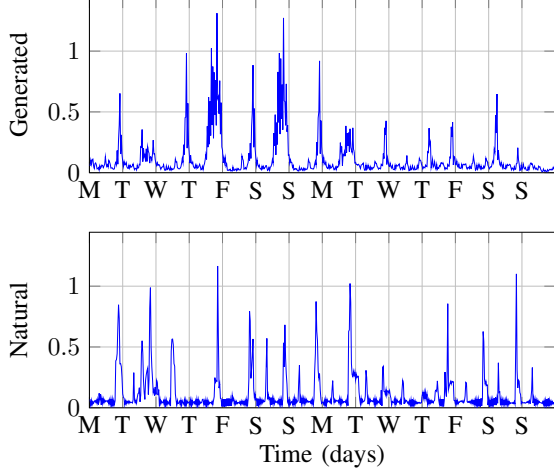


Fig. 2: Two generated weeks of power consumption data (in kWh), as well as the closest (in terms of Earth Mover's Distance) curve in the training set.

mechanisms and provide a 20% gain over a random guess, even in unfavorable conditions for the attacker. On the other hand, the gradient-norm attack can perform better in poor training conditions, but is easier to mitigate.

Fig. 1 shows predictions performed by an LSTM network on a curve from the test set. The LSTM model always predicts half a day (24 points) from the preceding week (336 point) of ground truth data at a time. The plot results from joining these predictions together. Fig. 2 showcases a random artificial power curve generated by the same LR generative model. Below it is the closest natural curve from the model's training set, in terms of their EMD.

IV. SUMMARY AND CONCLUDING REMARKS

We provided a privacy-aware method for power consumption data generation. Additionally, we proposed two novel methods for evaluation of GANs for sequential data. Firstly, the LSTM score, which uses the forecasting performance of an LSTM and can be generalized to all types of sequential data. Secondly, the Average Indicator Distance which uses the statistics of power consumption curves. To this end, we chose features that are useful for classification tasks on power consumption data [2], [13].

We studied the interplay between privacy and quality in generative networks for power consumption data. We found that there is a trade-off between the quality of the generated samples and the privacy guarantees of the generative model. When using the discriminator for attacks, the success rates are similar to those found for classifiers [16], [18], [19]. Yet, with proper training setup, the model can be robust against membership inference attacks, which provides a strong guarantee against other attacks, like attribute inference.

Our model can be improved and adapted to other types of power consumption data. For example, by taking readings over different time spans or conditioning on

weather/temperature. Our method for data generation and privacy analysis allows power companies to share power consumption data in a useful and privacy-preserving way.

ACKNOWLEDGMENT

This research was supported by the Gaspard Monge program for Optimization and Operations Research (PGMO). This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 797805.

REFERENCES

- [1] SmartMeter Energy Consumption Data in London Households - London Datastore. <https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households>.
- [2] Christian Beckel, Leyna Sadamori, and Silvia Santini. Towards automatic classification of private households using electricity consumption data. pages 169–176, 11 2012.
- [3] Ali Borji. Pros and Cons of GAN Evaluation Measures. *arXiv:1802.03446 [cs]*, Feb. 2018. arXiv: 1802.03446.
- [4] Yize Chen, Yushi Tan, and Deepjyoti Deka. Is machine learning in power systems vulnerable?, 2018.
- [5] Yize Chen, Yishen Wang, Daniel Kirschen, and Baosen Zhang. Model-free renewable scenario generation using generative adversarial networks, 2017.
- [6] M Rodríguez Fernández, I González Alonso, and E Zalama Casanova. Online identification of appliances from power consumption data collected by smart meters. *Pattern Analysis and Applications*, 19(2):463–473, 2016.
- [7] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *ACM Conference on Computer and Communications Security*, 2015.
- [8] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. pages 17–32, 2014.
- [9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]*, June 2014. arXiv: 1406.2661.
- [10] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. A review on generative adversarial networks: Algorithms, theory, and applications, 2020.
- [11] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. LOGAN: Membership Inference Attacks Against Generative Models. *Proceedings on Privacy Enhancing Technologies*, 2019(1):133–152, Jan. 2019.
- [12] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, Nov. 1997.
- [13] Konstantin Hopf, Mariya Sodenkamp, Ilya Kozlovkiy, and Thorsten Staake. Feature extraction and filtering for household classification based on smart electricity meter data. 11 2014.
- [14] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, Dec. 2014. arXiv: 1412.6980.
- [15] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral Normalization for Generative Adversarial Networks. *arXiv:1802.05957 [cs, stat]*, Feb. 2018. arXiv: 1802.05957.
- [16] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive Privacy Analysis of Deep Learning: Stand-alone and Federated Learning under Passive and Active White-box Inference Attacks. *arXiv:1812.00910 [cs, stat]*, Dec. 2018. arXiv: 1812.00910.
- [17] Alex Sherstinsky. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *arXiv:1808.03314 [cs, stat]*, Aug. 2018. arXiv: 1808.03314.
- [18] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks against Machine Learning Models. *arXiv:1610.05820 [cs, stat]*, Oct. 2016. arXiv: 1610.05820.
- [19] Stacey Truex, Ling Liu, Mehmet Emre Gursay, Lei Yu, and Wenqi Wei. Towards Demystifying Membership Inference Attacks. *arXiv:1807.09173 [cs]*, June 2018. arXiv: 1807.09173.