GAN-based Model for Residential Load Generation Considering Typical Consumption Patterns

Conference Paper · November 2018

DOI: 10.1109/ISGT.2019.8791575

5 authors, including:

Yuxuan Gu
Tsinghua University

Qixin Chen
Tsinghua University

Chongqing Kang
Tsinghua University

GAN-based Model for Residential Load Generation Considering Typical Consumption Patterns

Yuxuan Gu*, Qixin Chen*, Kai Liu†, Le Xie‡ and Chongqing Kang*
*Department of Electrical Engineering
Tsinghua University, Beijing, China 100084
Email: qxchen@tsinghua.edu.cn
†Southern Power Grid Dispatch Center, Guangzhou, China 510623
‡Department of Electrical and Computer Engineering
Texas A&M University, College Station, TX USA 77843-3128

***Abstract*—With the fast development of the Energy Internet, collecting and analyzing loads from the residential sector has become more and more important. However, the large-scale and real-time collection of residential loads still remains a big challenge due to high cost, technical barriers, and privacy concerns. Previous researches proposed two approaches for generating residential load profiles: bottom-up and top-down. However, these approaches suffer from either high complexity or low accuracy. In this work, we propose a residential load profiles generation model based on the Generative Adversarial Network (GAN). The GAN includes two independent networks: the generator and the discriminator which are trained against each other until achieving balance. Then we can produce synthetic profiles with the trained generator. To capture underlying features in load profiles, We transform them into matrices and implement convolution layers in networks. Furthermore, considering residents have different typical load patterns, We propose an advanced GAN based on the Auxiliary Classifier GAN (ACGAN) to generate profiles under typical modes. We use K-means clustering to acquire the pattern class of load profiles and train the model with the labeled data. Case studies on the dataset from an Irish smart meter trial show that the model can generate realistic load profiles under different load patterns without loss of diversity.**

***Index Terms*—Generative Adversarial Networks, load generation, load profiles, residential load, deep convolution networks**

I. Introduction

The residential load is an important part of the whole electricity consumption in society. It supports researches in various areas on the demand side such as customer characterization [1], direct thermostat control [2] and so on which have been reviewed in [3]. However, residents load is difficult to collect. First, smart meters have not been widely deployed in some areas owing to high cost. Second, the collected data have not been explored thoroughly because of technical barriers to measuring, storing and communicating. Third, people concern more about privacy and have promoted promulgation of related laws in recent years. Under this circumstance, generating artificial load profiles in accordance with real ones has become an innovative research idea and caused widespread concern among the scholars. The generated profiles can be used to explore the diversity and stochastic behind residential loads which helps promote the planning, construction, and management of terminal distribution networks.

Existing load generation techniques can be concluded as two approaches: bottom-up and top-down. Top-down approaches are less encountered in the literature comparing with the bottom-up. They take the residential load as a sink and do not consider the details of the individual end-users [4]. O'Neal regarded residential load as a function of macroeconomic factors, energy price and general climate [5]. Labeeuw proposed a top-down approach based on Mixture Model Clustering and Markov Models [6]. The method got around the privacy problem and allowed for Monte Carlo simulations. Since top-down approaches only need aggregate data, they are relatively easy to model. However, low modeling cost leads to the loss of diversity and accuracy.

Bottom-up approaches try to figure out the contribution of each end-user towards the aggregate house load. They start with independent individuals or appliances in the household. Subbiah proposed a generation method based on individual-level energy consuming activities concerning interactions of household members [7]. It was a novel way to generate realistic and high-resolution load profiles. Farzan built a model which can adapt to the penetration of Plug-in Electrical Vehicles (PEV), smart devices or new Demand Side Management (DSM) [8]. Ding present a model considering the electrical models of system components and random usage patterns of home appliances [9]. Marszal-Pomianowska proposed a high-resolution model using the 1-min cycle power characteristics of independent appliances as the main modeling block [10]. Though bottom-up approaches can generate realistic load profiles with diversity, they have more strict requirement on detailed information of the household and higher modeling cost. As mentioned above, previous methods have limitations either on accuracy or on complexity. Thus, a model with less dependence on the input data, lower modeling cost, and higher accuracy is required. Fortunately, the Generative Adversarial Network (GAN) inspires us.

This paper presents a residential load profiles generation model based on the GAN. It is a novel method to learn the underlying distribution of the given data, which has been used in renewable scenario generation already and performed well [11]. To capture abstract features behind the profiles, we implement convolution layers in the networks. To consider different load patterns of residents, we build an advanced model based on Auxiliary Classifier GAN (ACGAN).

The contributions of this paper can be summarized as followed: First, we incorporate the GAN model with load generation for the first time. Second, we improve the GAN to generate profiles under typical load patterns of residents. Third, the model performs well on the data from an Irish smart meter trial with regard to both diversity and similarity.

The rest of this paper is organized as followed: Section II presents the framework of the GAN. Section III presents the implementation of the proposed method. Section IV shows experimental results using load data from an Irish smart meter trial. Section V draws the conclusions.

II. Load Profile Generation Model

A. Generative Adversarial Network

The Generative Adversarial Network (GAN) was first proposed by Ian Goodfellow in 2014 [12]. It is a novel method to learn the underlying distribution of the given data. Because of its excellent performance, it has been widely used to produce synthetic samples when researchers are in lack of real samples or want to explore more diversity in real samples.

The GAN has two components: the generator and the discriminator as shown in Fig. 1. They are trained against each other. The goal of the generator is to generate synthetic samples to fool the discriminator while the discriminator aims to distinguish between the real and generated samples. Let G denote the function executed by the generator and D denote the function executed by the discriminator. The generator is fed with a noise vector Z under a known distribution (e.g. Joint Gaussian) denoted as $Z \sim \mathbb{P}_z$. The discriminator is fed with real samples X under the distribution $\mathbb{P}_x$ denoted as $X \sim \mathbb{P}_x$ or generated samples $G(Z)$ alternately. The output of the discriminator is denoted as $D(X)$ or $D(G(Z))$ which indicates the probability that the sample is real.

During every epoch of training, the generator is fed with a batch of noise vectors and the discriminator is fed with a batch of real samples X or generated samples $G(Z)$. We denote the average of the discriminator output in every batch as $\mathbb{E}$. Then two networks update their weights according to the descent of the loss function which can be formulated as followed:

$$Loss_G = -\mathbb{E}_{Z\sim\mathbb{P}_z}[D(G(Z))] \quad (1)$$

$$Loss_D = -\mathbb{E}_{X\sim\mathbb{P}_x}[D(X)] + \mathbb{E}_{Z\sim\mathbb{P}_z}[D(G(Z))] \quad (2)$$

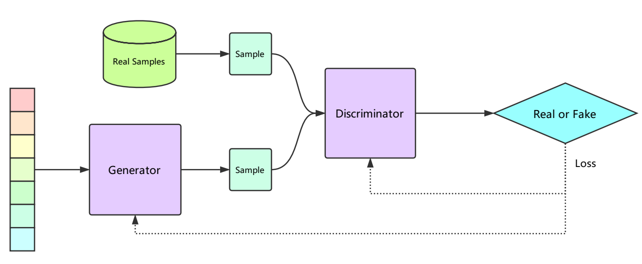

Fig. 1. The structure of the GAN

In the training process, the discriminator updates weights to maximize the distance between $D(G(Z))$ and $D(X)$ which means it is able to tell the difference between real and fake samples. The generator updates weights to maximize $D(G(z))$ which means the discriminator mistakenly take input samples as real ones. After the training finished, two networks reach Nash equilibrium. Then we can use the generator to produce synthetic samples.

B. Proposed Convolution GAN Model

Although the GAN performs well in many tasks such as high-resolution image generation and image sharpening, there still remain many problems. First, the GAN is hard to train because we should balance the evolution of two networks carefully in case one network is much stronger than the other. In practice, the training of the GAN is unstable and requires a bunch of hacks to work. These hacks have been summarized by Radford in 2016 as Deep Convolutional GAN [13]. Second, we cannot control the generated samples in the original GAN because the input is only composed of noise. Too high model freedom leads to the loss of accuracy.

To solve these problems, we propose an advanced GAN in this work. Our improvements include three parts: First, implementing convolution layers to extract underlying features. Since loads at the adjacent time in one day or the same time on adjacent days have the strong correlation, we transform weekly load profiles into load matrices whose rows are daily load profiles in one week. Then each element in the matrix is related to its adjacent elements just like pix-elated pictures. For the task of analyzing pix-elated pictures, convolution layers are the best choice. Second, several practical skills have been used to stabilize and accelerate the training process. We use batch normalization layers to clip the tensors in the networks to avoid vanishing or exploding gradient problem. We apply Relu activation function in the generator and leaky Relu activation function in the discriminator to accelerate the convergence of the training. Third, we propose an advanced GAN based on the ACGAN [14] to generate load profiles under typical load patterns which are corresponding to different consumption modes of residents. In the ACGAN, the generator receives both noise and labels as input, the discriminator exports not only real or fake probability but also category of the input sample, the loss function is composed of the likelihood of judging the authenticity and the category of the input together. We can control the type of generating load profiles through the label input of the generator. The details of the model are demonstrated as followed:

1) Generator: The generator receives noise vectors together with load pattern labels in one-hot encoding as input and exports generated profiles. First, we use a fully-connected layer to map noise vectors to the feature space. Second, two transposed convolution layers with stride size of 2×2 and kernel size of 4×4 are used to do up-sampling from the feature space to profiles. The size of the stride and kernel is determined by experiments. To stable the training process and help gradients flow through the nets, we use batch normalization layer after each computational layer. It modifies the output with zero mean and standard variation to help the tensors be activated equally in every dimension. The last transposed convolution layer does not need normalization to avoid concussion during training. In order to shape generated loads in the same interval as normalized real loads, we use $Sigmoid$ to activate the final output.

2) Discriminator: The discriminator receives real or generated samples as input and exports its judgment on the authenticity and category of the input. The architecture of the discriminator is symmetrical to the generator. First, two convolution layers are used to extract underlying features in load profiles. The convolution stride size and kernel size are the same as mentioned above. Second, we implement a fully-connected layer to extract the features like in the generator. Third, we design a binary-tree structure in the tail of the network since the output of the discriminator not only contains real or fake probability but also the category of the sample in the ACGAN. We implement one fully-connected layer mapping to the probability and two layers mapping to the category in one-hot encoding. The discriminator also needs batch normalization except the first layer and the last layer. After normalization, we use leaky Relu as the activation function.

Set the length of the noise vector at 90, the shape of weekly load matrices at 7×48 and the number of load pattern categories at 10. Then the GAN architecture can be designed as shown in Table I.

TABLE I
THE ARCHITECTURE OF THE PROPOSED MODEL

Layer	Layer Type	Hyper-parameters
	Generator	
G1	Fully Connected	Input size: 1×100 Neurons: 1920
G2	Transposed Convolution	Input size: $2\times12\times80$ Kernels: 256
G3	Transposed Convolution	Input size: $4\times24\times256$ Kernels: 1
	Discriminator	
D1	Convolution	Input size: $7\times48\times1$ Kernels: 256
D2	Convolution	Input size: $4\times24\times256$ Kernels: 256
D3	Fully Connected	Input size: 1×6144 Neurons: 3072
	Real or Fake	
D4	Fully Connected	Input size: 1×3072 Neurons: 1
	Category	
D5	Fully Connected	Input size: 1×3072 Neurons: 1536
D6	Fully Connected	Input size: 1×1536 Neurons: 10

III. IMPLEMENTATION AND EVALUATION CRITERIA

A. Implementation

The implementation of load profiles generation can be summarized as two parts: data preprocessing and model training. Data preprocessing includes data cleaning, normalization, data transformation and K-means clustering. Training process includes alternate weights updating between the discriminator and the generator.

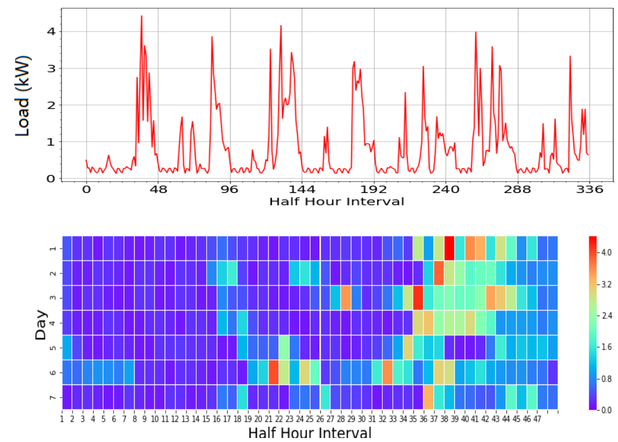

Fig. 2. Transformation from the weekly load profile to the load matrix

First, we need to remove the missing or abnormal data from the whole load set. Second, we need to normalize the data since the activation function in neural networks have good nonlinearity when its input in the interval $[0, 1]$. Third, we transform weekly load profiles with $7N$ points to $7 \times N$ load matrices whose rows are daily load profiles in one week in order to make full use of convolution layers in the networks. Randomly choose one profile from the whole set and its corresponding matrix is shown in Fig. 2. We can observe correlations among adjacent load pixels like loads at the same time on adjacent days are similar, load peak and valley occur at similar time points every day, and so on. However, there still remains many underlying features that cannot be observed by eyes which needs neural networks to find. Finally, we use K-means clustering to acquire typical load patterns as labels of massive profiles. The distance between two profiles in clustering is defined by Euclidean distance because each dim in the load vector is under the same scale.

The training process of the GAN is different from normal neural networks because we have two adversarial networks training alternately. Each net updates its weights based on the gradient descent when the other one is fixed. In practice, we set the learning rate of the generator 20 times of the discriminator, every time the generator is updated, the discriminator has enough steps to move to the optimal point. When we observe the convergence in the loss of the two nets, we can finish training and generate synthetic samples.

B. Evaluation Criteria

To evaluate the performance of the model, we proposed individual and aggregate criteria to validate the rationality of generated load profiles.

1) Individual Criteria: First, the generated load profiles should have randomness, volatility, and periodicity like real ones. Thus, we view weekly load profiles as time series and calculate its auto-correlation. Valley value in auto-correlation profiles means randomness while peak value means periodicity. Let us denote a load profile as $\{x\}_{t=1}^{N}$, then the auto-correlation value with lag h can be calculated as followed:

$$Autocorr(h) = \frac{1}{N-h} \sum_{t=1}^{N-h} \frac{(x_t - \hat{\mu})(x_{t+h} - \hat{\mu})}{\hat{\sigma}^2} \quad (3)$$

In the formula above, $\hat{\mu}$ and $\hat{\sigma}$ are the mean and standard variation of $\{x\}_{t=1}^{N}$. Second, generated profiles should have diversity to reflect all possible consumption curves of residents. The diversity can be observed by the scatter plot of the mean and standard variance of each generated profile. The more dispersed the scatter plot is, the more diversity generated profiles have.

2) Aggregate Criteria: Generated load profiles should have similar aggregate characteristics to real ones. To verify the similarity, we calculate the Root Mean Square Error (RMSE) between the mean profile of generated samples and real samples for each category. Let us denote the generated mean profile as $\{X_{gen}\}_{t=1}^{N}$ and the real as $\{X_{real}\}_{t=1}^{N}$, then their RMSE can be calculated as followed:

$$RMSE(X_{real}, X_{gen}) = \sqrt{\frac{1}{N} \sum_{t=1}^{N} (X_{real}^{t} - X_{gen}^{t})^2} \quad (4)$$

On the other hand, we calculate the Jenson-Shannon distance(J-S) between the discrete distribution of generated loads and real loads as followed:

$$JS(P \parallel Q) = \frac{1}{2} \sum_{x \in X} [P(x) \log \frac{2P(x)}{P(x)+Q(x)} + Q(x) \log \frac{2Q(x)}{P(x)+Q(x)}] \quad (5)$$

In the formula above, $P(x)$ and $Q(x)$ represent the probability of the generated or real load x in the whole interval X. Smaller RMSE and J-S imply higher accuracy of the model.

IV. Case Studies

A. Dataset Description

The dataset for the case study is from the Smart Metering Electricity Customer Behavior Trials [15] took place from 14/07/2009/ to 31/12/2010 with over 5000 Irish homes and businesses participating. We randomly choose 500 homes of all participants with 33760 half-hour weekly load profiles in total.

B. Results

Since residents have different load patterns, we use K-means clustering to acquire 10 typical load patterns as labels for massive profiles. These patterns have different amplitudes and shapes as shown in Fig. 3.

After data cleaning and normalization, we set hyper-parameters of the training process as followed: the learning rate of the discriminator at 0.00005 and that of the generator at 0.001. Two nets are trained alternately between one step of D and one step of G using Adam optimizer with a mini-batch size 0f 32. All weights for neurons in neural networks are initialized from a centered Normal distribution with the standard deviation of 0.02.

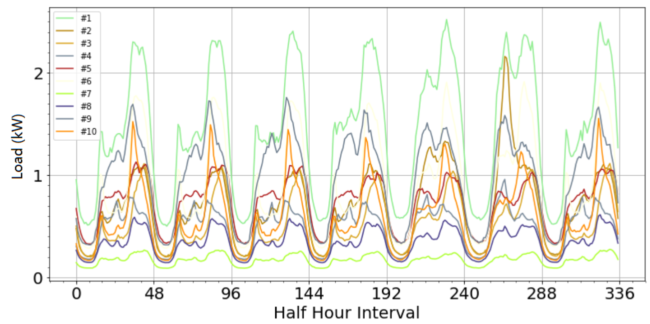

Fig. 3. Load profile centers of 10 categories

After the convergence of two networks, we use the generator to produce synthetic profiles. The aggregate criteria for each class are shown in Table II. Comparing to the average load level, the RMSE is very small. The J-S of each class is under 0.01 which represents the distributions of real and generated loads are quite similar. For a more intuitive understanding of the values, we plot the mean profile of generated and real samples for cluster 1,2,3,6,10 and the probability distribution function of generated and real loads for cluster 4,5,7,8,9 as shown in Fig. 4 and Fig. 5. It can be observed that two mean profiles and the probability distribution function of generated and real loads in each class are close to each other.

Four generated and real profiles are randomly chosen for validation of individual criteria. Their auto-correlation profiles are shown in Fig. 6. We can observe valley and peak values at adjacent points in auto-correlation profiles of generated and real samples. It indicates that both two profiles have random-

TABLE II
The RMSE and J-S Distance of 10 Categories

Cluster	RMSE/kW	J-S	Cluster	RMSE/kW	J-S
1	0.1616	0.001599	6	0.1409	0.001954
2	0.1216	0.002212	7	0.03316	0.002303
3	0.08304	0.002193	8	0.06151	0.003412
4	0.07690	0.002554	9	0.1519	0.002204
5	0.09112	0.001154	10	0.09389	0.003083

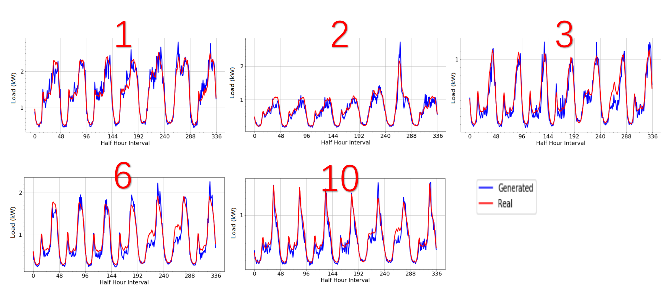

Fig. 4. Generated and real mean load profile of 5 categories

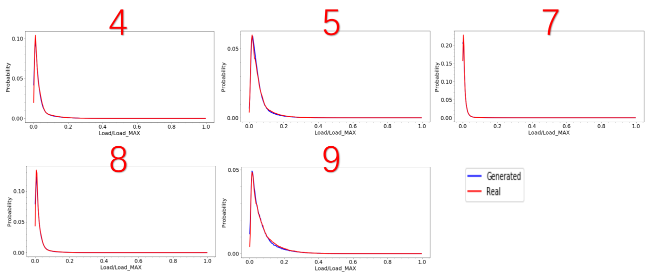

Fig. 5. Generated and real load probability distribution function of 5 categories

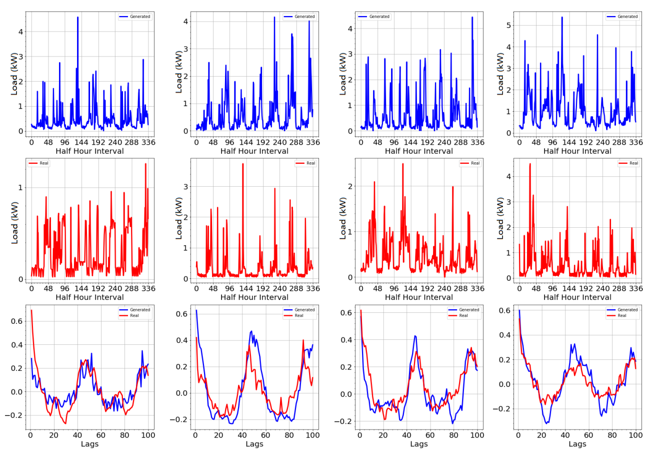

Fig. 6. Auto-correlation comparison between real and generated load profiles

ness and are in a cycle of one day (lags=48). Furthermore, to prevent the model from generating profiles under similar modes and losing the diversity, we draw the scatter plot of the mean and standard variance of each profile as shown in Fig. 7. The dispersed scatter plot of generated loads similar to real

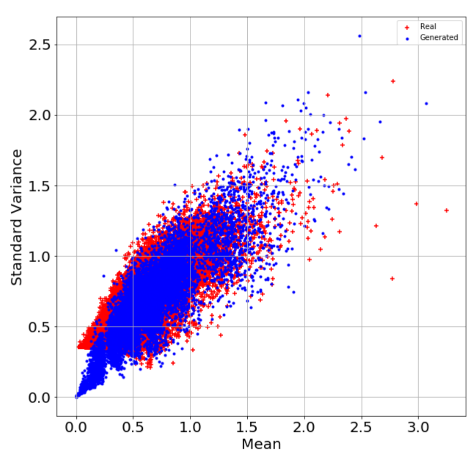

Fig. 7. Mean and standard variation of real and generated load profiles

ones represents that the generator can simulate various possible consumption profiles of residents like in the real world.

V. Conclusions

With the development of the Energy Internet, obtaining and analyzing load data from the residential sector is indispensable in many research areas on the demand side. However, due to the constraints on techniques and laws, a large-scale and real-time collection of residential load data still remains a big problem. Furthermore, generation of residential load profiles is a challenging problem because of high randomness, diversity, and volatility. This paper proposes a top-down approach to generate household load profiles under typical load patterns based on the GAN model. Experimental results show that the generator can produce synthetic profiles whose statistic characteristics are close to real ones and can reflect various consumption possibilities of residents. Through load profiles generation, we can explore the diversity and stochastic behind residential loads which helps promote the planning, construction, and management of distribution networks.

References

[1] Y. Wang, Q. Chen, D. Gan, J. Yang, D. S. Kirschen, and C. Kang, "Deep learning-based socio-demographic information identification from smart meter data," *IEEE Trans. Smart Grid*, 2018.

[2] S. Chen, Q. Chen, and Y. Xu, "Strategic bidding and compensation mechanism for a load aggregator with direct thermostat control capabilities," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 2327–2336, May 2018.

[3] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Transactions on Smart Grid*, 2018.

[4] L. G. Swan and V. I. Ugursal, "Modeling of end-use energy consumption in the residential sector: A review of modeling techniques," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 8, pp. 1819 – 1835, 2009.

[5] D.L.O'Neal and E.Hirst, "An energy use model of the residential sector," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 10, no. 11, pp. 749–755, 1980.

[6] W. Labeeuw and G. Deconinck, "Residential electrical load model based on mixture model clustering and markov models," *IEEE Trans. Industrial Informatics*, vol. 9, no. 3, pp. 1561–1569, 2013.

[7] R. Subbiah, K. Lum, A. Marathe, and M. Marathe, "Activity based energy demand modeling for residential buildings," in *2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*, 2013, pp. 1–6.

[8] F. Farzan, M. A. Jafari, J. Gong, F. Farzan, and A. Stryker, "A multi-scale adaptive model of residential energy demand," *Applied Energy*, vol. 150, pp. 258 – 273, 2015.

[9] T. Ding, H. Liang, and W. Xu, "An analytical method for probabilistic modeling of the steady-state behavior of secondary residential system," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2575–2584, 2017.

[10] A. Marszal-Pomianowska, P. Heiselberg, and O. K. Larsen, "Household electricity demand profiles a high-resolution load model to facilitate modelling of energy flexible buildings," *Energy*, vol. 103, pp. 487 – 501, 2016.

[11] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, "Model-free renewable scenario generation using generative adversarial networks," *IEEE Trans. Power Systems*, vol. 33, no. 3, pp. 3265–3275, 2018.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," *ArXiv e-prints*, Jun. 2014.

[13] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *ArXiv e-prints*, Nov. 2015.

[14] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *ArXiv e-prints*, Nov. 2014.

[15] Irish Social Science Data Archive, "Commission for energy regulation (CER) smart metering project." *www.ucd.ie/issda/CER-electricity*, 2012.