SGAN: Appliance signatures data generation for NILM applications using GANs

Christina Gkoutroumpi, Nikolaos Virtsionis Gkalinikis, and Dimitrios Vrakas

School of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece,
https://www.csd.auth.gr/en/

Abstract. The development and evolution of advanced energy system technologies is one of the most important goals for the global community in recent years. In this effort, the utilization and analysis of energy time series is of decisive importance for the understanding of energy consumption and production patterns. However, access to real data may be limited due to the sensitivity of the information and the limited amount of data already available. This has led to the use of methods to produce artificial data in order to enrich existing datasets. Generative Adversarial Networks or GANs are an approach to generative modeling using deep learning methods based on the logic of adversarial learning, and consist of two adversarial neural networks, a generator and a discriminator, which work together to produce realistic and unbiased data. The subject of the current paper is the creation of a GAN pipeline capable of producing power time series that resemble those observed in the real world, preserving the main characteristics and diversity of the observed electrical devices. The proposed method shows promising results, outperforming other state-of-the-art models in two calculated metrics.

Keywords: Synthetic Data, Deep Learning, Energy, Timeseries, GAN, NILM, Generative AI

1 Introduction

The term synthetic data, also known as artificial data, refers to information that has been generated using algorithms rather than from real-world monitoring. Synthetic data is used in cases where the volume of available data does not satisfy the requested requirements. Some of the most common techniques for generating artificial data include using deep learning and neural networks. The generation of synthetic data has been a key line of research in recent years, supporting a multitude of applications in computer science and artificial intelligence. The ever-increasing demand for large and diverse datasets, encompassing the range of complexity seen in the real world, is the main motivation behind the creation of synthetic data.

Using synthetic data has many benefits. Firstly, it can solve the problem of lack of data, as collecting and processing real data is an expensive and time-consuming process. Synthetic data also allows solving privacy problems, as real user information is not used directly. Data owners cannot share their data without safeguards in place. Legal concerns aside, there is a general reluctance to share sensitive data with non-experts before they have proven themselves trustworthy [1]. Finally, use of synthetic data allows control over the degree of complexity and parameters of the final data generated. Artificial data can be used to facilitate collaborations involving sensitive data. A good synthetic dataset has two properties: it is representative of the original data, and it provides strong privacy guarantees. Furthermore, to demonstrate that a method is robust to biased data, synthetic datasets containing appropriate 'corner cases' can be created. Bias checking of decision-making systems is especially important in cases where a 'black box' algorithm is used [2, 3]. In cases such as medical applications, data acquisition is a slow and expensive process, because usually the collection and interpretation of data requires the involvement of highly trained experts. To address such limitations, data augmentation, a set of techniques for increasing the size of a data set without collecting and annotating additional real data, can be performed [4].

Time series data, also known as dated data, is a series of data points that are organized in chronological order. Typically, these data points consist of consecutive measurements taken from the same source at fixed time intervals and are used to track changes that occur over time [5]. In recent years, several datasets of energy time series have been created, at different levels of sampling and amount of information. Some are available for free to the public [6, 7], while others are available for research purposes or through payment [8]. The primary objective of using synthetic data, in this context, is to employ it in the creation and evaluation of algorithms for Non-Intrusive Load Monitoring (NILM).

Non-intrusive load monitoring is a process which, by analyzing the changes in voltage and current of an electrical installation, infers which devices are being used at a given time as well as the energy consumption of each one of them [9]. NILM is considered a low-cost alternative to placing separate meters for each individual device. Detailed analysis of energy consumption in real time can lead to savings of up to 20% in energy consumption through detection of faulty devices and poor electrical management strategies [10]. Thus, in the long term, unnecessary waste of energy will be avoided, positively affecting the global warming and the climate change problems.

The contribution of the current paper to the field research of synthetic data for energy disaggregation could be summarized in three key points. To begin with, providing a pipeline for synthetic data generation where multiple appliances' generation can be accomplished through individual electrical appliance signal generations.The novelty is the proposed appliance signature preprocessing. A common practice involves directly feeding the entire time series data into neural networks and relying on the network to autonomously manage the learning process. In the presented approach, the network's ability to learn signatures more efficiently is facilitated by providing as input only the segments of the time series that contain the most critical information. This targeted input strategy enhances the learning process, enabling the network to discern relevant patterns more quickly. Moreover, a GAN architecture is proposed using convolutional layers, leveraging technics used for synthetic image generation. A novel metric called "normalized total energy" or NTE is also proposed to assess the synthetic data created from the neural networks. Lastly, a benchmark with 5 appliances between existing networks and comparison of the proposed method is created, while the code is provided on GitHub. The exploration of the proposed model extends to often overlooked use cases, specifically addressing scenarios such as Electric Vehicles (EV) and Air Conditioning (AC) signatures. These devices typically have limited available data but hold significant importance in the context of Non-Intrusive Load Monitoring (NILM).

The anatomy of this paper is as follows. For starters, there is a brief presentation of the related work on GANs, NILM and related datasets. Secondly, the data analysis and preprocessing process is analyzed. Section 5 contains information on the GAN topologies created and compared. The most important of the results are presented in section 6. Finally, conclusions and outlines of proposals for future work are presented.

2 Related Work

Generative Adversarial Networks (GANs) are an approach to generative modeling using deep learning methods. In a GAN network, two neural networks compete against each other in the form of a zero-sum game, where one agent's gain corresponds to another agent's loss. This concept was developed by Ian Goodfellow and his colleagues in 2014 [11]. The problem is framed as a supervised learning problem with two networks know as the *Generator* and the *Discriminator*. The generator model is trained to generate new realistic signals by taking random noise as input, and the discriminator model tries to classify signals as real (from the application domain) or fake (created by the generator).

GANs train the two networks alternately, completing one or more training epochs for one network and then switching to the opposing network for a similar duration. This process is repeated until the training process is completed. The weights of one network are held constant during the training of the opposing network. Otherwise, the generator would behave as if it were trying to achieve an ever-changing target and may never converge. The GAN game reaches equilibrium when the generator can create signals that are indistinguishable from real ones, rendering even a flawless discriminator unable to differentiate between them. For a GAN, convergence is often a fleeting, rather than a stable, state.

To enhance both the ultimate result and the stability of GAN training, numerous adaptations such as label smoothing, historical averaging, and minibatch discrimination have been introduced [12]. Another technique that was proposed by Metz et al. [13] is Unrolled GANs that use a generator loss function that incorporates not only the outputs of the current discriminator network, but also the outputs of future versions of it. This way, the generator cannot over-optimize for a single discriminator network. Arjovsky et al. [14] examines usual problems such as instability and saturation and introduces solutions. In [15] a new regularization approach with low computational cost is proposed to achieve a stable GAN training process. An algorithm named WGAN, an alternative to traditional GAN training, is proposed in [16]. They demonstrated that the KL divergence between the discriminator's outputs for real and fake samples, a frequently employed loss function in GAN training, faced issues with vanishing gradients. Wasserstein distance is also recommended as an alternative in [17].

It's worth mentioning that the generator architecture employed in GANs shows little substantial divergence from alternative methods, such as Variational Autoencoders [18]. In the context of VAEs, one network focuses on discovering more effective methods for encoding raw data into a lower-dimensional space, while the second network, known as the decoder, seeks to convert these representations into new data. Another variation is Progressive GAN, where the first layers of the generator generate low-resolution samples, while the subsequent layers gradually add finer details. This approach enables GANs to train faster than comparable non-progressive networks and produce higher resolution samples [19, 20]. Conditional Gans [21] are conditionally trained on a labeled dataset, and allow a label to be specified for each new sample generated.

The first to propose neural networks designed explicitly for Non-Intrusive Load Monitoring (NILM) were Kelly and Knottenbelt [22]. Numerous techniques have been proposed to achieve this goal, ranging from conventional signal processing approaches to advanced engineering algorithms rooted in machine learning [23–27]. In [28] the authors aim is to consolidate the autoencoder and GAN architectures into a unified framework, in which the autoencoder achieves a non-linear separation of the power signal source. A generalizable energy disaggregation pipeline is proposed in [29] and aims to address both the performance and efficiency aspects of Non-Intrusive Load Monitoring (NILM) models.

3 Datasets

While the generation of labeled real-world data for NILM applications is a labor-intensive and expensive endeavor, there are publicly available datasets that could be used of training machine learning models. The selected datasets include UK-DALE [30], a dataset consisting of 5 UK households, providing high frequency current and voltage measurements sampling. Another dataset used is REFIT [31], which includes full load recording of household activity and measurements of 9 individual devices, at 8 second intervals per house, collected continuously over a period of two years from 20 houses. Also, appliances from Pecan Street [8] were utilized, one of the largest and well-known datasets used for NILM procedures, offering data from both residential and commercial sources, gathered by the Pecan Street Research Institute from Austin, Texas, USA. These encompass various types of appliances commonly found in households, including washing machines, dishwashers, air conditioners and kettles.

Table 1. Table of selected appliances and their sources.

Appliance	Dataset	Country
Washing Machine	REFIT, UK-DALE, Dataport	UK, USA
Dishwasher	REFIT, UK-DALE	UK
Aircon	Dataport	USA
Kettle	REFIT, UK-DALE	UK
Electric car	Dataport	USA

4 Data Preparation

In this chapter, a detailed explanation is provided regarding the steps and methodologies involved in the analysis and processing of the data. The available time series data from the datasets used, are initially separated into 6 hour windows. This length was chosen as it allows to easily create synthetic data for a whole day by creating 4 such windows. The technique applied here was to isolate the device signature by pre-processing the time series, and then produce the final synthetic data in two phases. Essentially, this separates the data generation problem into two sub-problems. One problem is the shape of a device's signature (duration, number of pulses, pulse format). The second problem is when the device will operate during the day. It can be observed that these are two independent problems, since the signature of a device does not change with the time of day. A similar method is used in [32] where a synthetic dataset is created, by combining parts of observed appliances collected by them. Thus, the problem can be divided into two phases to solve it. This also allows for multiple devices to be easily added, which can be generated by different generator networks, while another network will coordinate when each device is turned on. This method also offers the ability to take into account the relationship between devices that tend to be used together or in a serial manner.

Following the generation of the relevant 6-hour windows from the input data, a series of data cleaning procedures is executed before proceeding to utilize and feed this data into the networks. The first step is to exclude the 6 hours that contain no appliance uses or contain exclusively noise. Windows with a very small energy sum are removed. In a next phase, the negative values that may be present in the samples, due to meter error, are removed. If any point is missing from a window, that window is excluded as well, due to missing values. Another step of the data preprocessing is the normalization of the values of the samples. This is done in order to help the network training and limit the range of possible values that the samples can have. For this transformation, the Min-Max normalization method is employed, and the features are rescaled to lie with the range of 0 to 1. The equation used can be seen in equation 1. Finally, in order to disregard samples containing a lot of noise, the variance of each time series is calculated and used to filter the samples further.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

To isolate the part of the time series containing the signature of the device from the rest of the series, appropriate pre-processing of the data is performed. More specifically, initially the point where the first use of the device begins in the specific six-hour sample is found, i.e. the first point with energy greater than a minimum limit. This was chosen so that very small meter noises would not be detected as the beginning of device use. Subsequently, the last point of device usage in the 6-hour window is detected. In order to determine whether the examined part of the time series requires further subdivision into independent usages, convolution is employed. More specifically, a window with uniform values of 1 is convolved with the selected part of the time series. Through this approach, the identification of the appropriate point for further division in the series is determined, if such a point exists. Then, each newly created series is analyzed with the same algorithm recursively. If a series subpart only contains noise, it is discarded and not considered further. This trimming step is repeated until there are no more splits to be done, or a specified number of repetitions is reached. This is how the different uses are separated, and any meter noise is removed, as a set of points containing noise will be isolated from the rest.

5 GAN topologies

5.1 GanNilm

One of the architectures implemented is that of GanNilm [33], which has been adapted to generate data for a single device. The original model proposed aimed to achieve non-intrusive load monitoring through the use of generative adversarial networks. Specifically, the generator network produces an output for each aggregate input measurement, which represents the detailed measurement of a device. This generated output is subsequently combined with the corresponding actual aggregate measurement and is fed into the discriminator network to determine if it is a real sample. To ensure network stability, the techniques of feature matching and output-input concatenation are employed. In this manner, the generator network learns how total consumption is distributed among individual devices, while the discriminator indirectly learns the loss function. Below the architecture of GanNilm is shown (see Fig. 1).

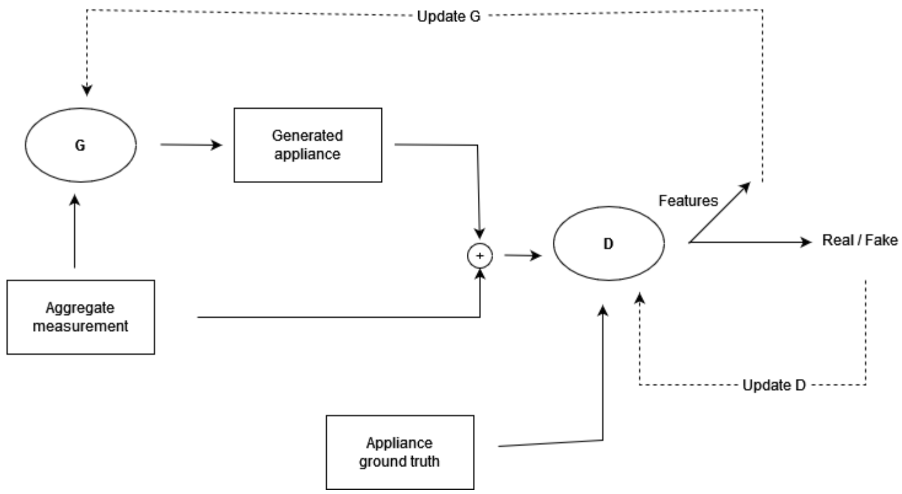

Fig. 1. The architecture of GanNilm network for performing NILM

5.2 DCGAN

The proposed architecture consists of two convolutional networks, a generator and a discriminator. The generator network consists of a series of convolutional layers at its initial stage. These layers take as input the random noise provided to the network and, through the use of activation function leaky ReLU, propagate the information through the network. Subsequently, the following layers are inverse convolutional layers, meaning they expand the size of the current time series generated. The final layers of this network consist of linear layers, which ultimately yield the desired time series length as the output. Dropout layers and batch normalization are used to increase training stability. Convolutional layers were chosen as they are fast, suitable for time series data, and effective at feature extraction. Batch normalization plays an important role in mitigating the internal covariate shift issue in deep neural networks. It standardizes the intermediate outputs of each layer within a batch during training, enhancing the stability and speed of the optimization process [34].

In Fig. 2 the architecture for the DCGAN Generator is shown, where the parameter k stands for the kernel size of the layer, ch is for the number of channels and s represents the stride used in the specific layer. The kernel size refers to the dimensions of the filter or convolutional kernel used during the convolution operation. In this case, where the convolution is done in 1 dimension, the size k dictates the width of the filter used. The stride is a parameter that dictates the movement of the kernel, across the input data, determining how many elements the convolutional filter moves at each step along the input sequence. Finally, the channels' parameter refers to the number of channels produced by the convolution operation.

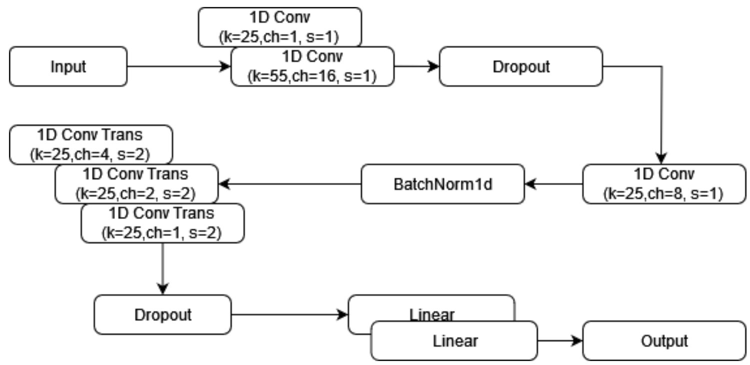

Fig. 2. The architecture of DCGAN Generator network

The Discriminator network consists of convolutional layers, interspersed with dropout layers. The final two layers are linear, utilizing the Sigmoid activation function to produce the final result. In Fig. 3 the architecture for the DCGAN Discriminator is shown.

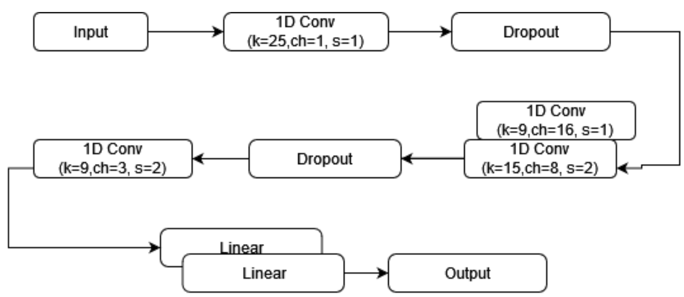

Fig. 3. The architecture of DCGAN Discriminator network

5.3 SGAN (Signature GAN)

A similar architecture as described above for DCGAN is employed in this model, with the difference that signature isolation as described in Data Preparation Chapter is used before feeding the data into the neural network. The GAN is trained on the signatures of each one of the appliances, while a new generator network is produced for each one of them. To generate synthetic data, the Generator produces a signature specific to the appliance. In the subsequent phase, the starting point of the generated appliance is established based on the specified 6-hour window that needs to be created. As the final step, this information is utilized to generate the complete time series.

In Fig. 4, the pipeline architecture for the SGAN network is shown. The initial step involves feeding the raw energy time series from the utilized datasets, mentioned in section 3, into the signature isolation preprocessing algorithm. This process aims to extract the specific appliance signature by employing the method detailed in section 4. This output is then used as ground truth for the discriminator during the training phase. As shown in Fig. 4 the discriminator network then is also utilized to update the generator network weights during the training phase. The generator network takes as input latent code, in order to produce realist appliance signatures based on its current training phase and the feedback from the discriminator. Once appliance signatures are generated, the subsequent stage involves finalizing the synthetic time series. In this step, the device operation starting time is defined, and the final time series is constructed using this information and the output of the generator, resulting in the final appliance operation window. It should be noted that the dashed lines in the figure denote operations exclusive to the training phase and are directed backwards.

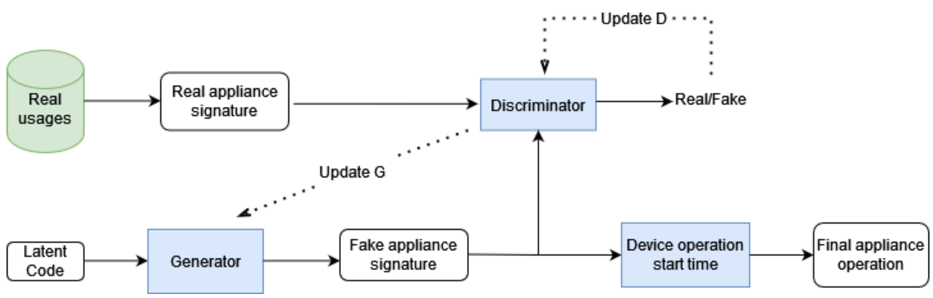

Fig. 4. The architecture of SGAN network for synthetic time series generation

The utilization of the signature isolation method in this architecture is demonstrated to enhance the final results of the GAN, as shown by the calculated metrics in the subsequent chapter. Despite sharing a similar architecture with DCGAN involving convolutional layers, this instance employs a reduced number of layers with fewer parameters. This not only contributes to an improvement in training efficiency by reducing the time required for model training but also results in a lighter neural network due to a smaller input size and reduced parameter count.

6 Experiments

This chapter presents results and analyzes the methods and models developed in this work. The key pieces of code were developed in the Python language version 3.10.12. To train the algorithms and perform the experiments, Google Colab-T4 GPU and Titan X Nvidia GPU were used. For the visualization of the results, WanDB 0.15.10 was utilized.

Three key evaluation methods are employed to assess the network's performance. The first method involves comparing the total energy predicted by the network to the total energy of the actual samples. The second technique involves an empirical analysis of the results. The third method utilizes the FID (Fréchet Inception Distance) score to measure the similarity of synthetic samples in comparison to real samples. The FID score [35] was created as a metric for assessing the similarity between two sets of image data. It has been demonstrated to have a strong correlation with human evaluations of visual quality, and is frequently employed to evaluate the quality of samples produced by Generative Adversarial Networks. FID is computed by measuring the Fréchet distance between two Gaussian distributions that are fitted to representations obtained from the Inception network. A smaller metric value indicates a superior evaluation of the synthetic network data in comparison to the real data.

The equation used for calculating the FID score is the following where (Mt,Ct) and (Mg,Cg) represent the mean and variance of real and synthetic features respectively:

$$FID = ||M_t - M_g||_2{}^2 + Tr(C_t + C_g - 2(C_t C_g)^{1/2}) \tag{2}$$

In table 2 the results for the fid score comparison between the real data and the generated data from the compared architectures are shown. It can be observed that for the 4 out of 5 appliances the proposed architecture achieves better results. Specifically, for the dishwasher and washing machine, the proposed method achieves 80% lower FID score compared to the GANNilm architecture. On the other hand, for the electric car GANNilm achieves 70% better FID score. For the aircon appliance, the biggest difference is observed, with the SGAN architecture achieving better results than the other networks.

Table 2. Results from fid score comparison between generated appliances.

fid score *10^{-3}			
appliance/network	SGAN	GANNilm	DCGAN
dishwasher	**0.0874**	0.22	31
washingmachine	**0.118**	0.3	0.2
aircon	**0.18**	3.6	4.4
kettle	**0.12**	0.92	0.77
electric car	1.3	**0.6**	1.2

Another metric employed is the "normalized total energy", which is calculated as the average total energy of the synthetic and real time series, after they have been normalized. This metric helps assess the energy distribution and characteristics of the data in both synthetic and real datasets.

Table 3 displays the percentage deviation in the average total energy between the real and synthetic data. This deviation is calculated separately for each method and for each device, providing insights into the differences in energy characteristics between the two types of data. A smaller deviation signifies a

Table 3. Results from normalized total energy comparison between generated appliances.

NTE perc diff %			
appliance/network	SGAN	GANNilm	DCGAN
dishwasher	**14.21**	65.52	221.11
washingmachine	**1.36**	85.07	31.61
aircon	28.21	**5.34**	45.09
kettle	**19.39**	121.02	82.13
electric car	**7.43**	17.81	56.46

better result, aligning with the findings of previous evaluation methods. As is evident from the visual representation of the results, models that introduce a significant amount of noise in their output tend to exhibit greater variability in the average energy of a time series. This additional noise contributes unwanted energy to the waveform. Models that produce less noise in their output tend to achieve a total energy of the time series that closely aligns with that of the real dataset. Of course, the total energy used can fluctuate depending on the duration of device usage.

For certain appliances like air conditioners, where the pulse duration is determined by the user, there can be significant variations in total energy across all the waveforms, even within the actual data. Moreover, for the washing machine, the proposed method achieves a much lower NTE score compared to the GAN-Nilm architecture and the DCGAN architecture. As the washing machine has a distinctive signature, the total energy of the synthetic time series achieves a close approximation of real-world data. More specifically, as it can be seen in table 3, the generated time series for washing machine and dishwasher have a low NTE score for SGAN achieving a deviation of only 1.4% and 14% when compared to the real time series from the utilized datasets. The higher deviations of 19% and 28% are observed for the kettle and aircon appliances, both of which have a total energy consumption which depends highly on each usage duration.

Figure 5 displays a sample of generated signals for each of the trained appliances from SGAN, alongside corresponding real power traces. It's evident that the generated signals exhibit highly comparable behaviors and contain all the key characteristics of each appliance category.

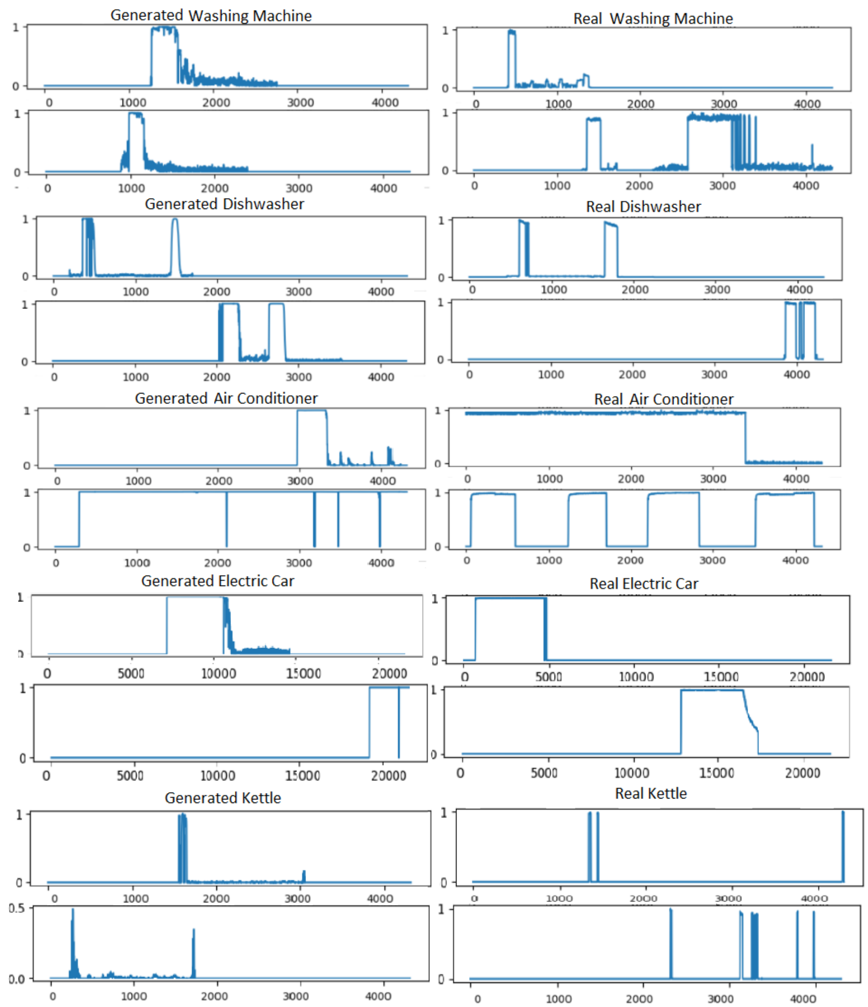

Fig. 5. Visual comparison of generated appliances (left column) and real appliances (right column) using the SGAN approach.

7 Conclusions and Proposals for Future Work

The presented paper incorporates a novel technique known as signature isolation into a neural network to improve its performance. Preliminary results showcase the capability of the introduced GAN architecture to generate synthetic energy data that closely mimics the characteristics of real-world energy consumption across a diverse range of appliances. It is observed that appliances exhibiting high variance in usage duration, show substantial disparities between usages and may impact the results of the calculated metrics.

Comparing the results of the proposed method to existing architectures leads to promising conclusions. Notably, devices with more distinct signatures, like washing machines and dishwashers, exhibit better results in both calculated metrics and visual assessments. Conversely, devices such as air conditioners or electric vehicle chargers, where pulse duration depends on user behavior, showcase greater diversity in consumed energy and signature shape. Experiments on a wider range of target devices may provide more insights on this topic.

Importantly, this method can be applied to generate synthetic data for an entire household by selecting the devices to be included and aggregating the results from each GAN generator for individual devices. For this purpose, a second network could be developed to facilitate the integration of generated signatures from multiple devices. This capability enables the generation of synthetic data for an entire household, giving users the flexibility to select the devices to be included in the aggregated dataset. Such a model could extend its input to not only include historical information about when a device starts operating, as currently implemented, but also consider multiple device activation times. This approach would leverage existing correlations between device startup times, such as those between a washer and dryer, to generate more realistic data. Furthermore, the technique of signature isolation could be applied to other networks with the goal of enhancing their performance.

References

1. Howe, B., Stoyanovich, J., Ping, H., Herman, B., Gee, M.: Synthetic data for social good. (2017). arXiv preprint arXiv:1710.08874
2. Soltana, G., Sabetzadeh, M., Briand, L. C.: Synthetic data generation for statistical testing. In 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE) (pp. 872-882). IEEE. (2017, October).
3. Quintana, Daniel S: A synthetic dataset primer for the biobehavioural sciences to promote reproducibility and hypothesis generation(2020-03-11)
4. Rajotte, J. F., Bergen, R., Buckeridge, D. L., El Emam, K., Ng, R., Strome, E.: Synthetic data as an enabler for machine learning applications in medicine. (2022). Iscience, 25(11).
5. Ralanamahatana, C. A., Lin, J., Gunopulos, D., Keogh, E., Vlachos, M., Das, G.: Mining time series data. Data mining and knowledge discovery handbook,(2005). 1069-1103.
6. Pereira, L., Quintal, F., Gonçalves, R., Nunes, N. J.: Sustdata: A public dataset for ict4s electric energy research. In ICT for sustainability 2014 (ICT4S-14) (pp. 359-368). Atlantis Press. (2014, August).
7. Kolter, J. Z., Johnson, M. J.: REDD: A public data set for energy disaggregation research. In Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA (Vol. 25, No. Citeseer, pp. 59-62). (2011, August).
8. Parson, Oliver, et al. "Dataport and NILMTK: A building data set designed for non-intrusive load monitoring." 2015 ieee global conference on signal and information processing (globalsip). IEEE, 2015.
9. Ruano, A., Hernandez, A., Ureña, J., Ruano, M., Garcia, J.: NILM techniques for intelligent home energy management and ambient assisted living: A review. Energies, 12(11), 2203. (2019).doi:10.3390/en12112203

10. Naghibi, B., Deilami, S.: Non-intrusive load monitoring and supplementary techniques for home energy management. In 2014 Australasian Universities Power Engineering Conference (AUPEC) (pp. 1-5). IEEE. (2014, September).
11. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems, 27.(2014).
12. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. Advances in neural information processing systems, 29.(2016).
13. Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J. : Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163. (2016).
14. Arjovsky, M., Bottou, L. : Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862 (2017).
15. Roth, K., Lucchi, A., Nowozin, S., Hofmann, T. : Stabilizing training of generative adversarial networks through regularization. Advances in neural information processing systems, 30. (2017).
16. Arjovsky, M., Chintala, S., Bottou, L.: . Wasserstein generative adversarial networks. In International conference on machine learning (pp. 214-223). PMLR. (2017, July)
17. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A. C. : Improved training of wasserstein gans. Advances in neural information processing systems, 30. (2017).
18. Kingma, D. P., Welling, M. : Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114. (2013).
19. Karras, T., Aila, T., Laine, S., Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196. (2017).
20. Harell, A., Jones, R., Makonin, S., Bajic, I. V. PowerGAN: Synthesizing appliance power signatures using generative adversarial networks. arXiv preprint arXiv:2007.13645. (2020).
21. Smith, K. E., Smith, A. O. Conditional GAN for timeseries generation. arXiv preprint arXiv:2006.16477. (2020).
22. Kelly, J.; Knottenbelt, W. Neural nilm: Deep neural networks applied to energy disaggregation. In Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments, Seoul, Republic of Korea, 4–5 November 2015; pp. 55–64.
23. Gomes, E., Pereira, L. : PB-NILM: Pinball guided deep non-intrusive load monitoring. IEEE Access, 8, 48386-48398.(2020)
24. Krystalakos, O., Nalmpantis, C., Vrakas, D. : Sliding window approach for online energy disaggregation using artificial neural networks. In Proceedings of the 10th Hellenic Conference on Artificial Intelligence (pp. 1-6). (2018, July).
25. Virtsionis-Gkalinikis, N., Nalmpantis, C. and Vrakas, D. SAED: self-attentive energy disaggregation. Mach Learn 112, 4081–4100 (2023). doi:10.1007/s10994-021-06106-3
26. Nalmpantis C, Virtsionis Gkalinikis N, Vrakas D. Neural Fourier Energy Disaggregation. Sensors (Basel). 2022 Jan 9;22(2):473. doi:10.3390/s22020473
27. Virtsionis Gkalinikis, N., Nalmpantis, C., Vrakas, D.: Variational Regression for Multi-Target Energy Disaggregation. Sensors, 23(4), 2051. (2023). doi:10.3390/s23042051
28. Kaselimi, M., Doulamis, N., Voulodimos, A., Doulamis, A., Protopapadakis, E.: EnerGAN++: A generative adversarial gated recurrent network for robust energy disaggregation. IEEE Open Journal of Signal Processing, 2, 1-16.(2020).

29. Dimitriadis I, Virtsionis Gkalinikis N, Gkiouzelis N, Vakali A, Athanasiadis C, Baslis C. : HeartDIS: A Generalizable End-to-End Energy Disaggregation Pipeline. Energies. 2023; 16(13):5115. doi:10.3390/en16135115
30. Kelly, J., Knottenbelt, W.: The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. Scientific data, 2(1), 1-14. (2015).
31. Murray, D., Stankovic, L., Stankovic, V.: An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study. Scientific data, 4(1), 1-12. (2017).
32. Klemenjak, C., Kovatsch, C., Herold, M., Elmenreich, W.: A synthetic energy dataset for non-intrusive load monitoring in households. Scientific data, (2020). 7(1), 108.
33. AHMED, Awadelrahman MA; ZHANG, Yan; ELIASSEN, Frank. Generative adversarial networks and transfer learning for non-intrusive load monitoring in smart grids. In: 2020 IEEE international conference on communications, control, and computing technologies for smart grids (SmartGridComm). IEEE, 2020. p. 1-7.
34. Kurach, K., Lučić, M., Zhai, X., Michalski, M., Gelly, S.: A large-scale study on regularization and normalization in GANs. In International conference on machine learning (pp. 3581-3590). PMLR. (2019, May).
35. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30. (2017).