Neural network to generate synthetic building electrical load profiles

Francesca Conselvan[1], Daniel Harringer[2], Daniele Antonucci[3] and Philipp Mascherbauer[2]
1 e-think energy research, Argentinierstrasse 18/10, 1040 Wien
2 Vienna University of Technology, Institute of Energy Systems and Electrical Drives, EEG TU Wien, Gusshausstrasse 27-29/373-2, A-1040 Vienna, Austria
3 Institute for Renewable Energy, Eurac Research, Viale Druso Drususallee, 1, 39100 Bolzano, Autonome Provinz Bozen - Südtirol, Italy

Abstract

Nowdays, we can collect a considerable amount of data in the energy sector, but, due to privacy concerns, companies are unable to share customer meter data for general research and analysis. Synthetic data provides a valid alternative to real data since they anonymize the data and maintain its original statistical information. This paper presents a real case scenario, where we generated 62 synthetic load profiles using the GAN algorithm, DoppelGANger. DoppelGANger specifically focuses on time-series data, even if they are multidimensional and have mixed categories. The proposed approach has three main steps: (1) preparing the data (2) clustering the load profiles with the k-means and the agglomerative algorithms (3) using DoppelGANger to generate synthetic load profiles.

Keywords Electrical load profiles, Generative Adversial Network (GAN), synthetic data, Machine Learning, clustering time/series data

1. Introduction

The increasing use of building monitoring and control systems has created an opportunity for data-driven approaches in the construction sector. Nevertheless, only 20% of buildings use up to 80% of the available building data. To change this trend, we need to openly share data to gain insights and make decisions. MODERATE aims to create an open-access platform of 50,000 buildings of different types (residential, commercial, offices, etc.) to bridge the gap between data and decision-making to reduce carbon emissions and mitigate climate change in the building sector. To comply with the General Data Protection Regulation and not share sensitive information, Machine Learning (ML) techniques are used to create synthetic data and offer access to building datasets. Synthetic data is artificially generated data that maintain the statistical characteristics of real and protects sensitive information by replacing identifiable information with fake data. Currently, the use of synthetic data is not widely applied in the construction industry and is one of the elements that allows open data sharing.

2. Method and results

2.1 Data source and data preparation

In this case study, we used 393 electrical load profiles with a temporal granularity of one hour for one year and a half. The load profiles came without metadata information, and we did not know if they are residential or non-residential buildings, so we did a clustering to have better insights of the energy usage. We first prepared the data for clustering by removing the missing data (52 profiles) and adding a layer of temporal granularity for a better classification of the data. We labelled the data points according to the season they have been recorded (winter, spring, summer, and autumn), if they fall at the weekend or not, and specified the month, the weekday, and the year of the record. Second, we normalized each load profile by its annual peak load to make each building similar in magnitude. Electrical load profiles can vary significantly according to customer type, customers' energy-usage patterns, climate conditions, etc. and that can influence the clustering.

2.2 Load profile clustering

Clustering is an unsupervised learning technique that organizes data into groups based on similarity and identifies trends and relationships between data points. In the area of electrical load profiles, clustering helps to identify patterns in energy consumption trends and informs, consequently, on the type of building. We tried two different clustering methods: k-means and agglomerative clustering. K-means is a clustering algorithm that uses a centroid-based approach to partition a data set into pre-set k clusters. The clusters iteratively assign each data point to the cluster with the nearest mean (centroid) until the cluster converges. For the k-mean, the first step was to select the optimal number of clusters and we used the GAP statistic algorithm[1]. Gap Statistics chooses the number of K according to the highest value based on the overall behaviour of drawn samples. As shown in Figure 1, the optimal is k=11.

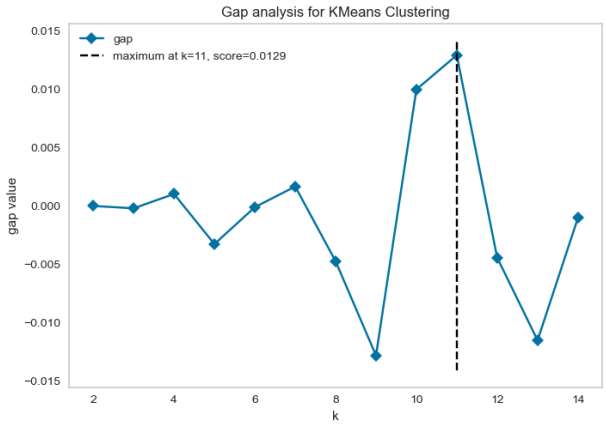
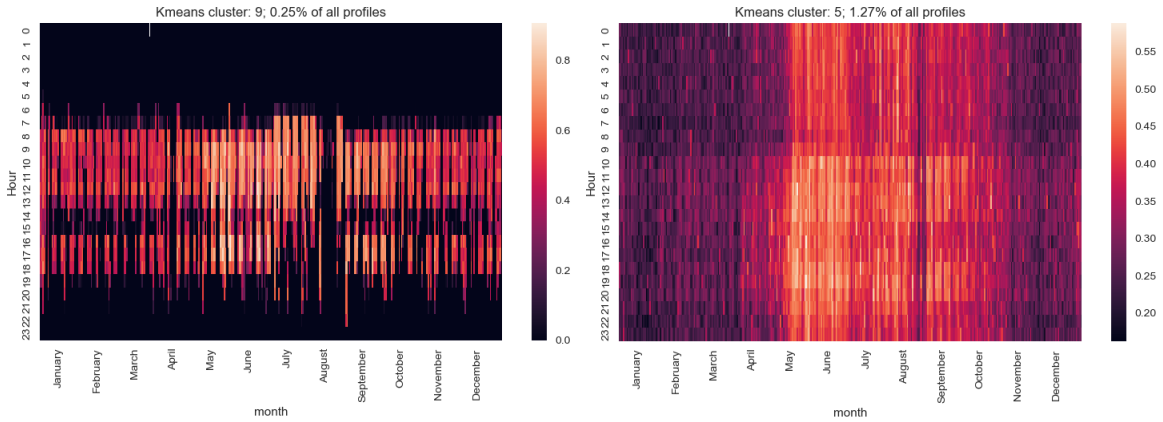


Figure 1: Number of clusters according to GAP analysis method

[1] (Tibshirani, Walther and Hastie, 2001)

Each cluster accounts for between 0.25% and 28% of the total daily load profiles. As shown in Figure 2, some clusters show a clear working day, while others a peak of energy use during summer.

Figure 2: k-means clustering

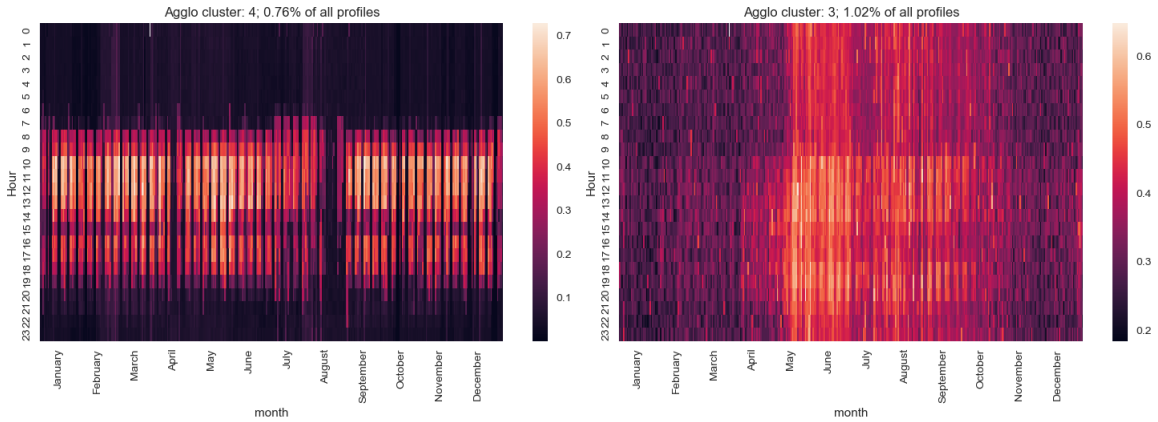


a) electricity usage increases in the morning and decreases in the evening

b) electricity usage increases during the summer period

Agglomerative clustering is the most popular type of hierarchical cluster, it has a bottom-up approach and, compared to k-means, does not require to have a pre-specified number of clusters. Each object is initially considered as a separate cluster and iteratively combined with other clusters in a hierarchical structure. The data points are organized into a tree-like structure (dendrogram), in which each node represents the entire dataset, the branch length represents the degree of dissimilarities between clusters and each branch represents a cluster (Figure 4). As Figure 3 shows the agglomerative method shows a similar energy usage trend of the k-means with peaks during the day or in summer.

Figure 3 Agglomerative clustering



a) electricity usage increases in the morning and decreases in the evening

b) electricity usage increases during the summer period

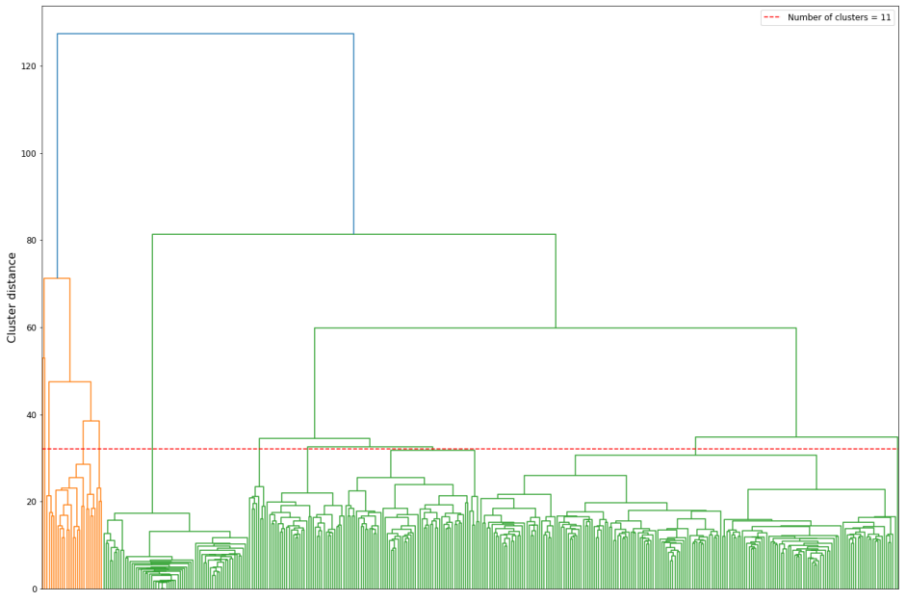


Figure 4 Dendrogram of agglomerative clustering

2.3 Generative Adversial Network (GAN) and data synthesis

GAN was first proposed by Goodfellow in 2014[2] and since its introduction, it has been widely used to generate new images, music, and data. GAN is a type of deep learning algorithm that uses two neural networks to generate new data. The two networks, the generator and the discriminator, compete with each other in a game of cat and mouse. The generator creates new data, while the discriminator determines whether the samples are real or fake. The ultimate goal is to create synthetic data that is indistinguishable from the real.

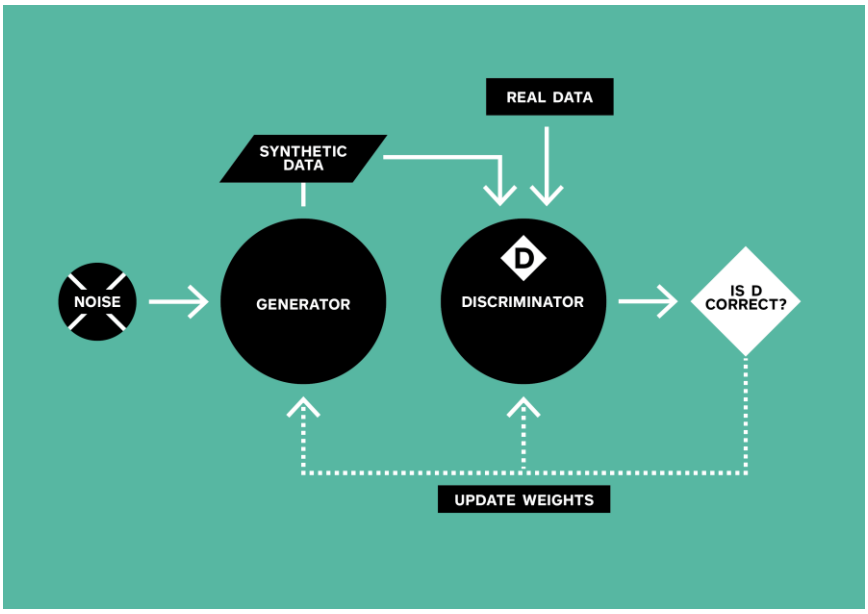


Figure 5: GAN workflow (picture by SquareUp)

One of the advantages of using GAN over other algorithms is that it can generate more realistic synthetic data, as it can capture the underlying distribution of the data. Moreover, GAN is less computationally expensive and generates data faster. Among the different algorithms listed as GAN, we opted for DoppelGANger.

DoppelGANGer is a method recently developed by researchers from Carnegie Mellon University and IBM[3] for sharing time series data with high fidelity and promising privacy properties. As a GAN, the model uses an adversial training scheme to better capture correlations between time series and their attributes, generates smaller stacked batches for long sequences, decouples normalization, supports fixed variables that do not change over time, and requires a small amount of the hyper-parameters. We used the implementation distributed by the company GretelAi[4]. Due to computational constriction, we tested the algorithm using a reduced dataset, specifically a meteorological season, Winter

[2] (Goodfellow *et al.*, 2014)

[3] (Lin *et al.*, 2020)

[4] https://github.com/gretelai/gretel-synthetics

(1.12.2021 - 28.2.2022) for a total of 90 days, and a cluster of 62 load profiles. In this way, we could understand the algorithm workflow and easily adjust the hyper-parameters to have good-quality synthetic data. The sample length is 129600 values and the max sequence length of 24 hours. We set the number of training samples used in one iteration of training (batch size) to 90 and the number of iterations (epochs) to 10000 training. The batch size controls the accuracy of the estimate of the error gradient when training neural networks, while in each epoch, the discriminator and generator take turns improving their parameters by learning from each other. The goal of each epoch is to improve the overall performance of the DoppelGANger. The learning rate is another important parameter that determines how much the weights of the network are adjusted in proportion to the calculated gradient. We tuned it to a relatively small value (1e-4). Smaller values lead to slower convergence, but may lead to better generalization. Larger values can lead to faster convergence but can also lead to overfitting. The loss function evaluates how well the algorithm models the dataset. A GAN model consists of two parts, one for the generator and one for the discriminator, and measures how well the generator is. We used the pre-set loss function Wasserstein, which is a type of distance metric that is better suited to capturing the structure of the data than traditional loss functions, such as the Mean Squared Error (MSE).

The goal of DoppelGANger is to reproduce load profiles, which have the same key statistical information as the real profiles. We adopted two approaches to validate whether DoppelGANger can generate load profiles that are similar to the real data. The first approach determines the degree of similarity between the profiles expressed by the autocorrelation score. The autocorrelation score shows the relationship of a time series at its present value as it relates to its previous value. In autocorrelation, the effects of the previous time lags are removed. Its value ranges from -1 (perfect negative correlation) to +1 (perfect positive correlation).

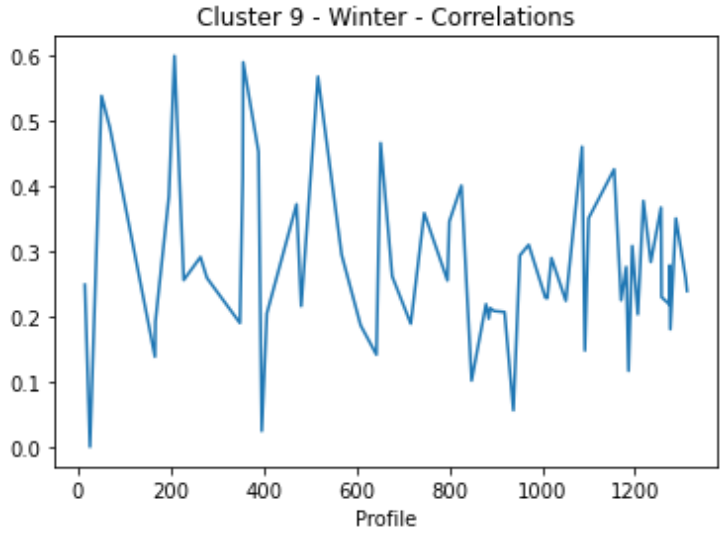


Figure 6: Autocorrelation analysis

The second approach compares the time-series distribution of the 62 load profiles of the synthetic to real data (Figure 7).

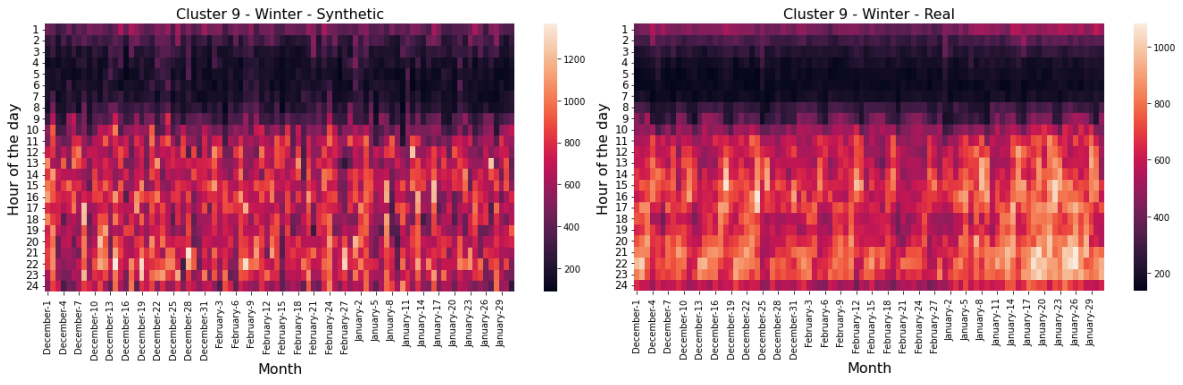


Figure 7: Agglomerative clustering of real data vs synthetic data

3. DISCUSSION

3.1 Future directions

Our work is still at an early stage and these are preliminary results, but already the work done so far has denoted the next steps to take:

1. Finding meaningful clusters using feature extractions[5]:

   We plan to engineer new features and use them instead of the original ones in the clustering procedure. Instead of using the hourly 'Consumed energy' values directly, we can compute matrix profiles and use them to derive vectors that store the (z-normalized) Euclidean distance between any subsequence within a time series and its nearest neighbour[6]. In this way, we would also add an extra layer of anonymization.

2. Generate high-quality synthetic data:

   Knowing the geographic location of the buildings, we can correlate the outside temperature to the energy used to generate synthetic data that more accurately reflects a home's energy usage. For example, if it is known that a home typically uses more energy when the outside temperature is warmer, this data can be used to create more realistic synthetic energy usage data.

3. Use additional methods to validate the fidelity of the synthetic data:

   We want to use additional metrics to validate the fidelity of the synthetic data, like a visual representation of the distribution of the key parameters. We will illustrate whether the cardinality and the ratio of the categories are respected. We also want to assess how well the

[5](Hennig *et al.*, 2020)

[6] https://stumpy.readthedocs.io/en/latest/Tutorial_The_Matrix_Profile.html

synthesized dataset can succeed in common data science problems. Measuring the performance of synthetic data can be done by testing different ML models, such as XGboost.

3.2 Future uses

The synthetic data will feed an open-access database part of an open platform of ML tools aimed to analyze the data for real-case scenarios. In system management, data can shed light on the performance of buildings, predict energy consumption and make diagnoses of energy systems. It can identify possible faults and ensure better operational efficiency by reducing energy consumption and improving occupant indoor comfort. For instance, ML can detect anomalous energy consumption behaviours either generated by end users, appliance failures, or other causes. A precise estimation of energy consumption can help to reduce energy waste and energy costs. That can help inform renovation roadmaps, and strategies for whole carbon reduction as well as define better-informed investment decision-making in new buildings.

4. CONCLUSION

Our proposed approach has three steps. We first normalized the load profiles to the max of the annual peak load to make sure they are similar in magnitude. Then we used the k-means, the agglomerative, and the hierarchical methods to cluster the data and check if they produce clusters with similar trends of energy usage. Lastly, we used DoppelGANger to generate the load profiles for a single cluster. We validated our method by comparing the time-series distribution and looking at the autocorrelation score between load profiles. The proposed approach can be used to generate building load profiles, anonymize smart meter data for sharing and support research and analysis to implement building energy efficiency.

ACKNOWLEDGEMENTS

Funded by the European Union

The research conducted in this paper has been funded by the European Union's Horizon Europe innovation program under Grant Agreement No. 101069834. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. For more information, see the project website: https://moderate-project.eu/

BIBLIOGRAPHY

Gretel.Ai, https://github.com/gretelai/gretel-synthetics

Goodfellow, I.J. *et al.* (2014) 'Generative Adversarial Networks'. arXiv. Available at: http://arxiv.org/abs/1406.2661 (Accessed: 30 January 2023).

Hennig, M. *et al.* (2020) 'Comparison of Time Series Clustering Algorithms for Machine State Detection', *Procedia CIRP*, 93, pp. 1352–1357. Available at: https://doi.org/10.1016/j.procir.2020.03.084.

Lin, Z. *et al.* (2020) 'Using GANs for Sharing Networked Time Series Data: Challenges, Initial Promise, and Open Questions', in *Proceedings of the ACM Internet Measurement Conference*, pp. 464–483. Available at: https://doi.org/10.1145/3419394.3423643.

STUMP, https://stumpy.readthedocs.io/en/latest/Tutorial_The_Matrix_Profile.html

Tibshirani, R., Walther, G. and Hastie, T. (2001) 'Estimating the number of clusters in a data set via the gap statistic', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2), pp. 411–423. Available at: https://doi.org/10.1111/1467-9868.00293.