D4.2 Model techniques for synthetic data creation

MODERATE

Marketable Open Data Solution for Optimized Building-related Energy Services

Table of Contents

Table of Contents 1
List of Figures 2
List of Tables 4
Executive Summary 6
Introduction 6
1 Load profiles on building level 7
1.1. Clustering algorithms 7
1.2. Literature review on synthetic data generation 13
1.2.1. Markov Chain (MC) 13
1.2.2. Neural Networks 14
1.2.3. Generative Adversial Network (GANs) 16
1.3. Generation of synthetic load profiles 19
2 Synthetic data generation for tabular data 24
2.1. Overview of the models 24
2.2. Datasets 26
2.3. Application on EPC Dataset of Lombardia Region 27
2.4. Application on LEIF Dataset 27
2.4.1. SYNTHESIS 27
2.4.2. DIAGNOSTIC REPORT 28
2.4.3. BOUNDARY ADHERANCE 29
2.5. Results 34
3 Machine Learning Techniques for BS Characterization 35
3.1. Synthetic Urban Building Energy Performance Data Generation 36
3.2. Building stock data disaggregation 37
3.3. HVAC identification 39
4 Building Stock Synthetic Data Generation: A Belgian Use Case 45
4.1. Current Building Stock 45
4.1.1. Time-series Electricity Consumption 45
4.1.2. Gas Consumption Profiles 47
4.2. Renovated Building Stock 49
4.3. Renovated Buildings with Heat Pumps 50

Funded by the European Union

Funded by the European Union. 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.

List of Figures

Figure 1: Elbow method distortion score for K-Means (left) and Agglomerative clustering (right) for cluster 2 to 15. 11
Figure 2: Silhouette score for K-Means (left) and Agglomerative clustering (right) for cluster 2 to 15. 11
Figure 3: Calinski Harabasz elbow score for K-Means (left) and Agglomerative clustering (right) for clusters 2 to 15. 12
Figure 4: Gap analysis for K-Means clustering for clusters 2 to 15. The optimal number of clusters is defined where the gap value reaches the highest point. 12
Figure 5: Davies Bouldin score for K-Means (left) and Agglomerative clustering (right) for clusters 2 to 15. 13
Figure 6: Visualization of a classic neural network. 14
Figure 7: Visualization of how a VAE works. 15
Figure 8: Visualization of the GAN principle. 17
Figure 9: Agglomerative clustering of real data vs synthetic data 20
Figure 10: energy consumption of a single load profile over one and half years (a) and for a single day (b) 21
Figure 11: synthetic load profiles. The red profiles are the synthetic load profiles and the grey the real data 21
Figure 12 Data Diagnostic Synthesis 28
Figure 13 Colum Coverage 29
Figure 14 Data Diagnostics: Column Boundaries 29
Figure 15 Distribution of real and syntethic data for column Country 30
Figure 16 Distribution of real and syntethic data for column building typology by Brick schema 31
Figure 17 Distribution of real vs synthetic data for “town” propriety of LEIF dataset 31
Figure 18 Distribution of real vs synthetic data for “energy consumption before refurbishment” propriety of LEIF dataset 32
Figure 19 Distribution of real vs synthetic data for “energy consumption after refurbishment ” propriety of LEIF dataset 32
Figure 20 Distribution of real vs synthetic data for “cost single project” propriety of LEIF dataset 32
Figure 21 Distribution of real vs synthetic data for “building area” propriety of LEIF dataset 33
Figure 22 List of refurbishment actions 33
Figure 23 Distribution of real vs synthetic data for “action value” propriety of LEIF dataset 34
Figure 24 The devised workflow to identify classifiers/clusters from building stock datasets using different data-drive machine learning techniques. 35
Figure 25 The simulation process workflow that uses a combination of data-driven and physics-based models to generate synthetic building energy performance data. 37
Figure 26 The disaggregation process workflow to generate decomposed end-use demand data. 38
Figure 27 High level flowchart of proposed multistep data enhancement methods for HVAC geospatial identification 39
Figure 28 Example of electricity consumption (from the dataset used in this study) for a building with and without heat pump. 40
Figure 29 Example of a comparison between load duration curves (with 15 mins time step) of the electricity consumption for two case studies with different HVAC installed. 40
Figure 30 Distribution of the different labels in the training and tests sets. 41
Figure 31 Sensitivity of the accuracy to the features and the ML technique. All the features from all months (with substantial missing points) were used in left. Only the second half of the year with complete datapoints in each month was used in training. 42

Funded by the European Union

Funded by the European Union. 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.

Figure 32 Confusion matrix for random forest method (y axis is the true label and x axis is the predicted label) 43
Figure 33 Feature importance for the developed model with random forest algorithm 44
Figure 34 Time-series electricity consumption profiles for the formulated building stock comprising 100 different building types with no renovations. 46
Figure 35 Electricity consumption classification of the building stock by month and by hour. 46
Figure 36 Feature importance of electricity time-series classifiers. 47
Figure 37 Time-series electricity prediction for the synthetic building stock using XGBoost algorithm. 47
Figure 38 Time-series gas consumption profiles for the synthetic building stock comprising 100 different building types with no renovations. 48
Figure 39 Gas consumption time-series classification of the building stock by month and by hour. 48
Figure 40 Feature importance of gas time-series classifiers. 49
Figure 41 Time-series gas prediction for the synthetic building stock using XGBoost algorithm. 49
Figure 42 Time-series electricity consumption profiles for the synthetic building stock comprising 100 different building types with fabric renovations, boiler upgrades and energy-efficient electrical appliances. 50
Figure 43 Time-series gas consumption profiles for the synthetic building stock comprising 100 different building types with fabric renovations, boiler upgrades and energy-efficient electrical appliances. 50
Figure 44 Time-series electricity consumption profiles for the synthetic building stock comprising 100 different building types with fabric renovations, heat pump installations and energy-efficient electrical appliances. 51
Figure 45 Electricity consumption time-series classification of the building stock by month and by hour. 51
Figure 46 Feature importance of electricity time-series classifiers. 52
Figure 47 Time-series electricity prediction for the synthetic building stock using XGBoost algorithm. 52

Funded by the European Union

Funded by the European Union. 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.

List of Tables

Table 1 Diagnostic Report for LEIF model 28
Table 2 Boundary adherence for LEIF dataset 30

Funded by the European Union

PROJECT DURATION: 1 July 2022 – 31 May 2026

WP4: DELIVERABLE: D4.2 Model techniques for synthetic data creation

LEAD BENEFICIARY: TUWIEN

SUBMISSION DATE: 31.12.2023

DISSEMINATION LEVEL: Public

DUE DATE: Draft version M12, final version M19

REVISION HISTORY:

DATE	VERSION	AUTHOR/CONTRIBUTOR[1]	REVISION BY[2]	COMMENTS
20/09/23	Draft	Philipp Mascherbauer - TUWIEN Francesca Conselvan - E-THINK Daniele Antonucci - EURAC Mohammad Haris Shamsi, Mohsen Sharifi, Yixiao Ma - VITO		
			Duncan Main - LINKS Yixiao Ma - VITO	VITO reviewed the work of TUWIEN, E-THINK and EURAC
29/01/24	Final	Philipp Mascherbauer - TUWIEN Francesca Conselvan - E-THINK Daniele Antonucci - EURAC Mohammad Haris Shamsi, Mohsen Sharifi, Yixiao Ma - VITO		
			Duncan Main - LINKS Yixiao Ma - VITO	VITO reviewed the work of TUWIEN, E-THINK and EURAC

Acknowledgements:

This project has received funding from the European Union's Horizon Europe research and innovation programme 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.

[1] Name SURNAME, ORGANIZATION

[2] Name SURNAME, ORGANIZATION

Funded by the European Union

Funded by the European Union. 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.

Executive Summary

The key goal of this deliverable is to describe the approach taken to create synthetic data that replicates real building data without infringing on data privacy concerns. Advantages and possible methods used to generate synthetic building data are explored within this report. Synthetic data has emerged as a powerful tool for data analytics and machine learning applications, primarily due to its data privacy compliance, high quality, and scalability. Adopting the appropriate synthetic data generation method is contingent upon the type and structure of the original data.

Introduction

In this deliverable, we explore the principal methods to generate synthetic building stock data and synthetic energy load profiles. The objective of the following chapters is to generate synthetic data, which resembles the original data, but does not violate any data security concerns[3]. Synthetic data offers several advantages at this extent:

- **Privacy**: Real data cannot always be used because they can contain sensitive information and violate privacy policies. Synthetic data are a valid alternative to real data, because it statistically resembles original data, but, at the same time, it protects personal information. The use of synthetic data for privacy protection is becoming increasingly popular due to its ability to mask personal information and prevent data breaches[4]. Synthetic data can be used to replace real data with artificial data that looks similar but does not contain any identifiable information. This protects individuals from data leakage and unauthorized access. In conclusion, synthetic data is a useful tool for organizations to meet the requirements of data privacy regulations such as the General Data Protection Regulation (GDPR)[5].
- **Quality**: Synthetic data ensures a high level of data, quality, balance and variety. Real data could be full of errors and inaccuracies that can affect the prediction models. Synthetic data fills the gaps, eliminates Inaccuracies and duplicates, enabling a more accurate use. The quality of synthetic data depends on the underlying algorithms and processes to generate it. To ensure the highest quality synthetic data, it is important to use reliable and robust methods that generate realistic data. It is also important to closely inspect the data and verify its accuracy before using it.
- **Scalability**: Machine learning models require a considerable amount of data to be properly trained. Synthetic data can easily supplement real world data for machine learning and data analysis. Synthetic data also has the advantage of being highly scalable, meaning it can be generated quickly and in large amounts. This makes it an ideal tool for large-scale data analysis and machine learning applications. Moreover, synthetic data can be tailored to the specific needs of the application, allowing great control and accuracy.

[3] T. E. Raghunathan, Synthetic data, Annual Review of Statistics and Its Application, 8, 129-140, 2021
[4] Jordon, James, and Jinsung Yoon. 'PATE-GAN: GENERATING SYNTHETIC DATA WITH DIFFERENTIAL PRIVACY GUARANTEES', 2019
[5] https://edps.europa.eu/press-publications/publications/techsonar/synthetic-data_en

Funded by the European Union

Funded by the European Union. 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.

1 Load profiles on building level

This chapter presents the reviewed literature and machine learning tools with the aim to generate synthetic data on a building level on an hourly/sub-hourly level. Generating data on an hourly level can be referred to as generating synthetic time series data. To generate synthetic time series data, specific models were developed throughout the years. In this chapter we list some common methods to generate synthetic time series data with their pros and cons and unsupervised machine learning algorithms to determine groups for a set of unlabeled load profiles.

1.1. Clustering algorithms

In order to create labels or to group the load profiles prior to creating synthetic data different clustering algorithms are applied based on the extracted features (see D4.16) from the profiles. Clustering is a pivotal step, especially when profiles are anonymized and cannot be classified according to their metadata information. For instance, attempting to train a model that amalgamates industrial and residential load profiles would result in the generation of erroneous synthetic data. To address this, we undertook an assessment of diverse clustering algorithms aimed at identifying analogous profiles and forming clusters. Through applying different clustering algorithms on different sets of features, we try to find the most suitable method of grouping unlabeled load profiles together.

In a first step we tried to cluster the normalized profiles without describing them through the extracted features using the K-Means, DTW, Hierarchical clustering and HDB-SCAN. Solely clustering the load profiles without any meta information proved to be very ineffective. First, the number of clusters is not consistent both between techniques and over time. Each method provides a different number of clusters, with certain techniques even yielding varying numbers with each run. Techniques to determine the number of clusters were the Elbow Method, Silhouette method, Calinski-Harabasz Index, Gap statistics, Davies-Bouldin index and the Dunn-index.

As a second step the clustering algorithms as well as the automatic number detection of a suitable number of clusters is done based on extracted features (see D4.1[13] chapter 2.3). These features capture statistics and properties of the load profiles, ideally capturing the behavior of the profile on different levels of aggregation. Simple examples of these features are the mean and the meadian on a daily, weekly or monthly basis. This section provides a comprehensive introduction of the primary clustering algorithms used and the methods used to determine the number of clusters.

Random forest (RF) is a machine-learning algorithm used for classification and regression. It is an ensemble method that combines the predictions of multiple individual models, the decision trees, to produce a more accurate and stable prediction. While random forest is very robust, less susceptible to overfitting and computationally very efficient, it also has some downsides specifically for our task.

Limitations:

- Random forest requires data being labeled for proper training. Nevertheless, the MODERATE partners will not necessarily provide the labels of the data, and this results in a clear limitation to have good quality results.
- Random forest can be sensitive to hyperparameters, which makes it difficult to achieve a consistent performance over varying input profiles. This means that a pre-trained model may

[6] Model techniques for synthetic data creation, Deliverable 4.1 Moderate project: https://moderate-project.eu/, 10.5281/zenodo.10534077

Funded by the European Union

Funded by the European Union. 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.

not deliver satisfactory results for new datasets which are uploaded to the MODERATE platform.

Nevertheless, this approach can yield satisfactory results. For instance, Yan et al. [7] use a random forest combined with a support vector machine to identify household characteristics from their respective load profiles.

K-Means[8,9] iteratively assigns each data point to the cluster, whose mean (i.e., centroid) is closest to the data point, and then update the centroids of the clusters based on the newly assigned data points. The process is repeated until the centroids no longer move significantly, or a maximum number of iterations is reached. The outcome of K-Means can differ since the first centroid is initialized randomly and K-Means needs to know beforehand how many clusters should exist within the dataset. This method is well-suited for clustering load profiles because it can identify clusters of similar load profiles and is relatively efficient.

Donaldson et al.[10] compared K-Means and hierarchical clustering to identify solar prosumers out of measured smart meter data. They state that dimension reduction of hourly profiles greatly reduces the amount of data while keeping a high degree of accuracy.

PAM-clustering stands for Partitioning Around Medoids and is closely related to the K-means clustering method. Instead of using the centroids as cluster centers it uses an actual datapoints which are called medoids. This makes it more robust to outliers compared to K-means and since the cluster is represented by a real datapoint the results can be more intuitive to interpret. Like K-means the number of clusters has to be determined beforehand and the result is dependent on the random initialization of the initial medoids. For very large datasets it might be less suitable than K-means as this algorithm is more computation intensive.

GMM-clustering[11] or Gaussian Mixture Models assume that the data is generated from several Gaussian distributions with a mean and a covariance. It tries to estimate these two parameters of each gaussian distribution updating them with each iteration until each data point reach their highest likelihood of being within a certain cluster. A big advantage of this approach over K-means is that it can deal with elliptical clusters as well, while K-means only identifies spherical ones. Additionally, each datapoint is given a likelihood of belonging to a certain cluster making it possible to have additional insights on the results. However, if the underlying data does not stem from Gaussian distributions, the algorithm is likely to perform poorly. Like K-means the number of clusters have to be determined beforehand and the result is dependent on the initial initialization, creating the necessity for multiple initializations methods.

Fuzzy C-means[12] is another clustering approach where each datapoint in each cluster receives a “membership value” between 0 and 1. 1 means that a data point belongs 100% to a certain cluster while 0 means that a data point is not part of a cluster. Like in K-Means the cluster centroids are

[7] S. Yan *et al*., "Time–Frequency Feature Combination Based Household Characteristic Identification Approach Using Smart Meter Data," in *IEEE Transactions on Industry Applications*, vol. 56, no. 3, pp. 2251-2262, May-June 2020, doi: 10.1109/TIA.2020.2981916

[8] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[9] https://tslearn.readthedocs.io/en/stable/gen_modules/clustering/tslearn.clustering.TimeSeriesKMeans.html

[10] https://doi.org/10.1016/j.ijepes.2020.105823

[11] https://scikit-learn.org/stable/modules/mixture.html

[12] https://pypi.org/project/fuzzy-c-means/

Funded by the European Union

Funded by the European Union. 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.

iteratively updated until they converge. Respect to K-means, Fuzzy C-means does not require the number of clusters beforehand however it is also computational more expensive.
It can be used to cluster load profiles, but due to its computational complexity, we cast it out from the options.

Hierarchical clustering[13] is an unsupervised learning algorithm that groups together similar load profiles based on the distance between data points. There are different options to calculate the distance and in time series, the Euclidian distance is the most used:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + \cdots + (d_n - d_m)^2}$$

The algorithm works by creating a hierarchy of clusters, which are progressively split into smaller and smaller sub-clusters, until a desired number of clusters is reached.

Hierarchical clustering can be further distinguished into two types:

- Agglomerative hierarchical clustering: this method treats every profile as a single cluster and successively merges the most similar profiles until a pre-defined number of clusters is reached.
- Divisive hierarchical clustering: this method first groups all the data points in a single cluster, and then it iteratively splits the cluster into smaller and smaller clusters until the desired number of clusters is reached.

DBSCAN[14] (Density-based spatial clustering) is usually used to identify dense clusters in a dataset. The main idea behind DBSCAN is to identify clusters by identifying areas of the dataset that are densely populated with data points, and then expanding these areas to include adjacent points that are also dense. Points that do not belong to a dense area are considered noise and are not included in any cluster. DBSCAN does not need an initial number of clusters, but rather a value (called epsilon or “eps”) that describes the maximum distance from a point to its neighbor. For each point in the dataset, all points within the distance of “eps” of this point are searched and added to a cluster. If certain points do not have any neighboring points within the pre-defined distance, they are considered outliers. Since we deal with load profiles, the choice of the correct “eps” value (maximum distance between points) is very hard and depends heavily on the input data. Depending on this value the algorithm can also find no cluster at all and define every point as an outlier or put all points into one cluster. The advantage is that it is not sensitive to the initial choice of centroids like K-Means, however, DBSCAN is not well-suited for very large datasets due to its computational complexity.

Like DBSCAN, **HDBSCAN**[15] identifies clusters by identifying areas of the dataset that are densely populated with data points and then expanding these areas to include adjacent points that are also dense. However, HDBSCAN also allows for the identification of clusters at different levels of granularity, by using a tree-based structure to represent the clusters. Like DBSCAN this algorithm relies on initial input on the expected distance between data points that should be in a cluster. These parameters are hard to estimate, and the results are solely dependent on them. Also, the load profiles which are labeled by the algorithm as “outliers” cannot be further used in our case, which makes DBSCAN and HDBSCAN not suitable for the synthetic load profile generation.

[13] https://github.com/scikit-learn
[14] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html
[15] https://github.com/scikit-learn-contrib/hdbscan

Funded by the European Union

Funded by the European Union. 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.

DTW[16,17] (Dynamic Time Warping) is a technique used to compare two time-series that may have different lengths and may be recorded at different times or under different conditions. Since DTW relies on an optimization algorithm that compares every time series with each other, it is computationally very expensive, and we consequently cast it out.

Mean-shift clustering is a centroid-based algorithm where each datapoint is shifted iteratively to the mean of its region (circular window) until it does not move anymore (or moves only very little). Datapoints having shifted to the same local maxima are considered to be within the same cluster. This clustering algorithm does not need a specified number of clusters beforehand and can detect clusters of any shape. However, due to the calculation of the mean of each datapoint within its region it can be very computationally intensive for large datasets. Additionally, it is sensitive to the bandwidth of the chosen window and the bandwidth cannot be chosen with a universal method.

Since it is possible that a user who wants to generate synthetic data, does not know how many inherently different groups of data are in the dataset, we investigated different methods to automatically determine the number of clusters.

The following methods are commonly used to automatically find the optimal number of clusters.

- Elbow Method
- Silhouette method
- Calinski-Harabasz Index
- Gap statistics
- Davies-Bouldin index
- Dunn-index

In the following we explain these methods with some exemplary results based on around 400 load profiles. The load profiles were obtained from a Spanish municipality and are completely anonymous. Only load values for each hour themselves and the maximum contracted power is known. Therefore these 400 profiles represent a variety of different small scale consumers which includes residential houses, small offices, bars, restaurants etc. The location of these consumers is not known, however for demonstrating the example results, no meta information on the profiles is relevant.

Elbow method is the most widely used method to determine the number of clusters. The clustering algorithm is run for a range of provided clusters and the within-cluster sum of squares is plotted for every number of cluster (Figure 1). The “elbow” in the plot represents the optimal number of cluster, meaning where the decrease in the within cluster sum of squares decreases significantly less with the next number of cluster. Figure 1 shows that 6 is the optimal number of clusters both for K-Means and Agglomerative methods.

[16] https://github.com/wannesm/dtaidistance

[17] https://tslearn.readthedocs.io/en/stable/gen_modules/clustering/tslearn.clustering.TimeSeriesKMeans.html

Funded by the European Union

Funded by the European Union. 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.

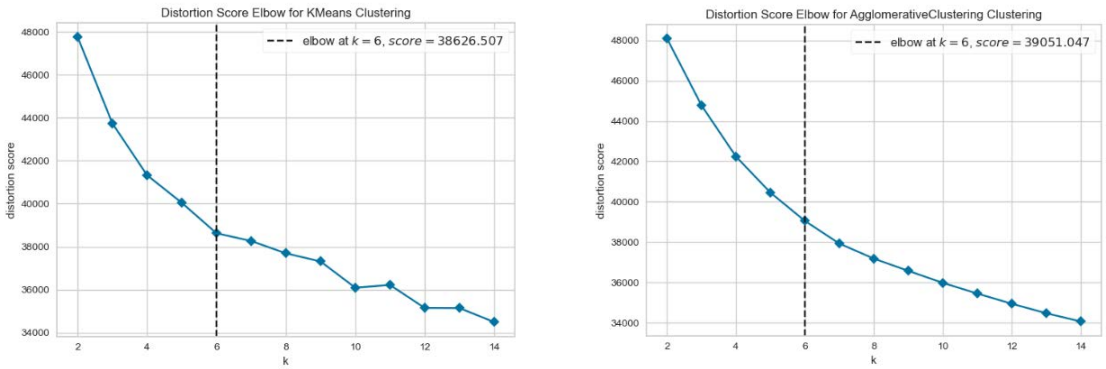


Figure 1: Elbow method distortion score for K-Means (left) and Agglomerative clustering (right) for cluster 2 to 15.

Silhouette method[18] compares the similarity of the data points within a cluster with the neighboring cluster. The score ranges from -1 to 1. A value close to 1 means that the distance of the samples within a cluster is much closer than to the neighboring cluster. 0 indicates overlapping cluster and a value close to -1 means that samples are assigned to the wrong cluster. For our case study, 2 is the optimal number of clusters (Figure 2).

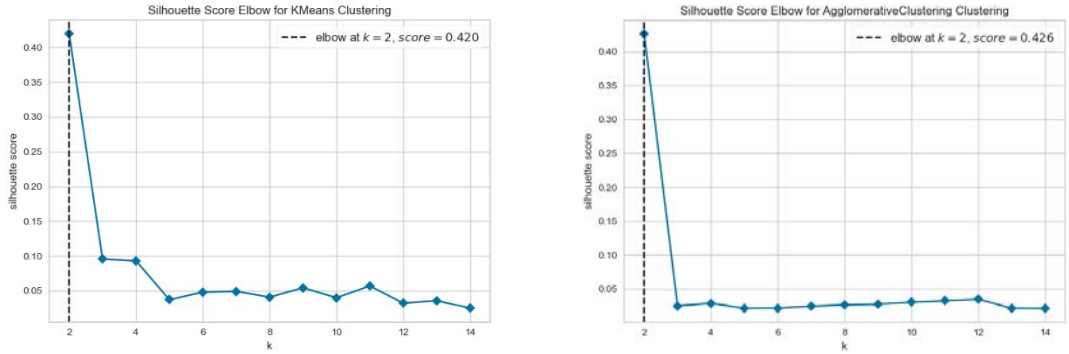


Figure 2: Silhouette score for K-Means (left) and Agglomerative clustering (right) for cluster 2 to 15.

Calinski-Harabasz Index[19] is calculated by dividing the variance of the sums of squares of the distances of individual objects to their cluster center by the sum of squares of the distance between the cluster centers. This provides information on how compact the clusters are within and how well spaced they are from different clusters. A high Calinski-Harabasz index corresponds to well-defined clusters. The Calinski-Harabasz index indicates that the optimal number of clusters in the data is two (Figure 3).

[18] https://github.com/scikit-learn/scikit-learn/blob/main/sklearn/metrics/cluster/_unsupervised.py
[19] https://github.com/scikit-learn/scikit-learn/blob/main/sklearn/metrics/cluster/_unsupervised.py

Funded by the European Union

Funded by the European Union. 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.

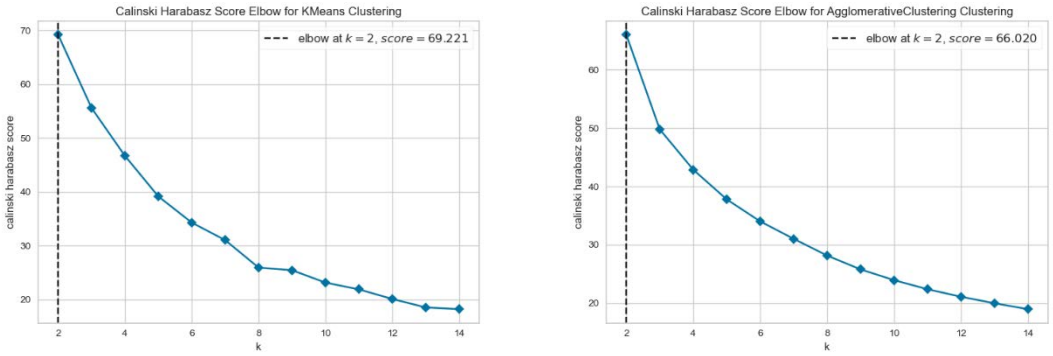


Figure 3: Calinski Harabasz elbow score for K-Means (left) and Agglomerative clustering (right) for clusters 2 to 15.

The concept behind **gap statistics** lies in the notion that the ideal cluster count corresponds to the -k value that maximizes the disparity ("gap") between the within-cluster sum of squares (WCSS) of the actual dataset and the corresponding WCSS calculated for a reference distribution. We calculated the reference distribution with K-Means by sampling the original data and randomly reassigning the observations to new clusters repeatedly for 10 times. The optimal number of clusters is the value where the gap is maximized. Because of the random initialization of clusters in the K-Means algorithm, the gap statistic can return a different optimal number of clusters for the same data. The maximum gap value is reached for 11 cluster, making 11 the optimal number of clusters (Figure 4).

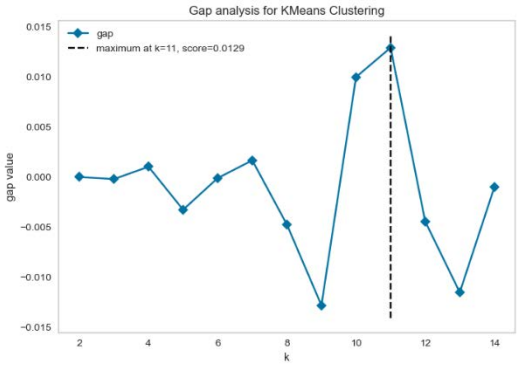


Figure 4: Gap analysis for K-Means clustering for clusters 2 to 15. The optimal number of clusters is defined where the gap value reaches the highest point.

The **Davies-Bouldin**[20] (DB) index is a measure of the compactness and separation of the clusters in a clustering algorithm. It is often used to evaluate the performance of a clustering algorithm, such as K-means. A lower DB index indicates better performance, with a value of 0 representing perfect separation and compactness. However, the DB index is sensitive to the number of clusters and may not always be a reliable measure of performance. Figure 5 illustrates the results for the DB index with 2 clusters reaching the lowest index.

[20] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies_bouldin_score.html

Funded by the European Union

Funded by the European Union. 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.

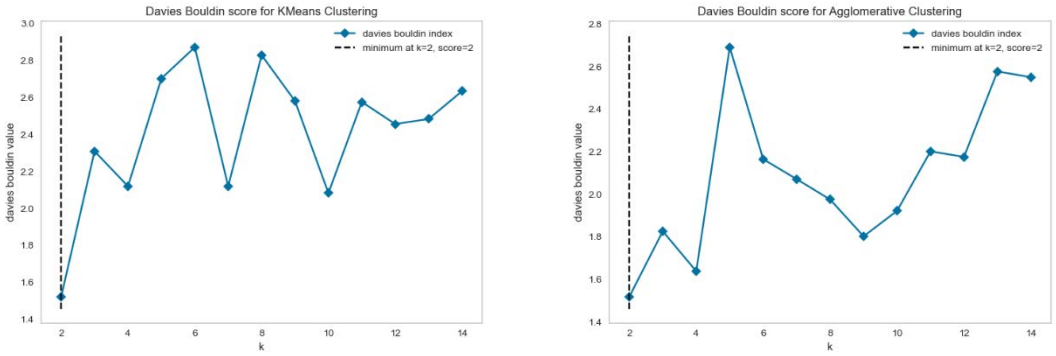


Figure 5: Davies Bouldin score for K-Means (left) and Agglomerative clustering (right) for clusters 2 to 15.

In general, the gap statistic has a much higher computational effort compared to the other above-mentioned methods. However, it should also be more reliable because the reference distribution is calculated numerous times to minimize the effect of the random initialization of the centroids.

The **Dunn-index** compares the compactness of a cluster to its separation from the other clusters:

$$Dunn\ Index = \frac{\min\ (inter\ cluster\ distances)}{\max\ (\text{intra cluster distances})}$$

The inter-cluster distances are the distances between clusters and it can be measured in various ways. For example, the distance between two centroids or the smallest distance between any pair of points from two clusters. The intra-cluster distances describe the spread of the cluster which can be described through the average distance of all points in the cluster from the centroid or by the distance of the furthest point of the centroid within a cluster. The Dunn-index is very intuitive and can be generally applied, however it is sensitive to outliers.

The cluster algorithms can be provided to the MODERATE platform if there is a need, and they could also find application in various other tasks involving categorical data. Users who want to generate synthetic load profiles must keep in mind that in order to get higher value results, the provided profiles should not be mixed at the beginning (e.g. Industry, residential buildings, service buildings).

1.2. Literature review on synthetic data generation

In this chapter we conducted a comprehensive analysis of existing literature to pinpoint the most suitable algorithms for producing synthetic load profile data. The generation process can be done using different machine learning (ML) and artificial intelligence (AI) techniques. In the following paragraph, we introduce the most common algorithms with particular focus on the generation of synthetic load profiles.

1.2.1. Markov Chain (MC)

Markov Chain is a mathematical system based on assumptions that activities evolve over time and future activities are only dependent on past activities. The transitions between states are determined by a probability distribution. The key property of a Markov chain is that the future state of the system can be determined from the current state and the transition probabilities, but not from the past states

Funded by the European Union

Funded by the European Union. 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.

or other external factors. Thus, Markov chains are often used to generate load profiles by describing the household's occupant behavior[21][22][23].

Advantages:

- Markov Chain can generate very realistic time series data as long as the underlying statistical data is correct.
- Markov Chain models are relatively simple and do not require high computational capacities.

Limitations:

- The Markov Chain is based on a Time Use Survey (TUS), since it relies on the underlying probability of every event happening throughout the day. TUS are surveys where people record their activities and the time they spend on them very precisely over a certain time period. With enough participants the TUS then delivers statistics on the behavior of people, their activities and time each activity needs. TUS are rare and none of the industrial partners provide such information, consequently discharging the use of this technique in the MODERATE project.

1.2.2. Neural Networks

Neural networks are a subset of machine learning and essentially are deep learning algorithms. A neural network consists of an input, an output and one or multiple hidden layers. Each layer consists of multiple nodes which have associated weights and are connected to each node of the neighboring layers. The weights of each node are updated during the training of the model until it provides satisfactory results.

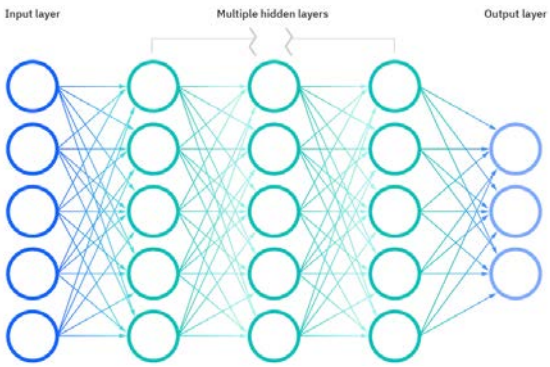


Figure 6: Visualization of a classic neural network[24].

[21] Bottaccioli, Lorenzo, Santa Di Cataldo, Andrea Acquaviva, and Edoardo Patti. 'Realistic Multi-Scale Modeling of Household Electricity Behaviors'. *IEEE Access* 7 (2019): 2467–89. https://doi.org/10.1109/ACCESS.2018.2886201;

[22] Ramírez-Mendiola, José Luis, Philipp Grünewald, and Nick Eyre. 'Residential Activity Pattern Modelling through Stochastic Chains of Variable Memory Length'. *Applied Energy* 237 (March 2019): 417–30. https://doi.org/10.1016/j.apenergy.2019.01.019;

[23] Flett, Graeme, and Nick Kelly. 'An Occupant-Differentiated, Higher-Order Markov Chain Method for Prediction of Domestic Occupancy'. *Energy and Buildings* 125 (August 2016): 219–30. https://doi.org/10.1016/j.enbuild.2016.05.015

[24] https://www.researchgate.net/publication/353234161_Wasserstein_GAN_Deep_Generation_applied_on_Bitcoins_financial_time_series

Funded by the European Union

Funded by the European Union. 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.

Neural networks can be classified into many different types. We only list three common ones which are also used for synthetic generation of time series data.

Feedforward network is a supervised learning algorithm that is used for classification and regression tasks and is trained to make predictions based on input data. For example, Gobind et al.[25] trained a neural network to generate regional load profiles based on the weather data as input.

Recurrent neural network is another type of artificial neural network. It is a dynamic model, which means that it can process and remember information from previous time steps, and can use this information to make better predictions at future time steps. Kleinebrham et al. combined a recurrent neural network approach with time of use survey (TUS) data to create a realistic household energy demand[26].

The third type of neural network that is investigated are the **Hybrid models.** This type of machine learning model allows to combine two or more different model architectures to create a single model. This can be done for a variety of reasons, such as to improve the accuracy of the model, to reduce the size of the model, or to make the model more versatile. We consider hybrid models to be too complex and computationally intensive to be considered as an option in the Moderate project.

Variational Auto-Encoder (VAE) is another artificial network widely used to generate synthetic data. It learns the distribution of an original dataset, encodes the data into the encoded or "latent" space and generates new data samples from the latent space with the decoder (see Figure 7). By minimizing the loss of information that happens when compressing and decompressing the data (training the model), we can optimize the model to generate data that is more like the original data[27].

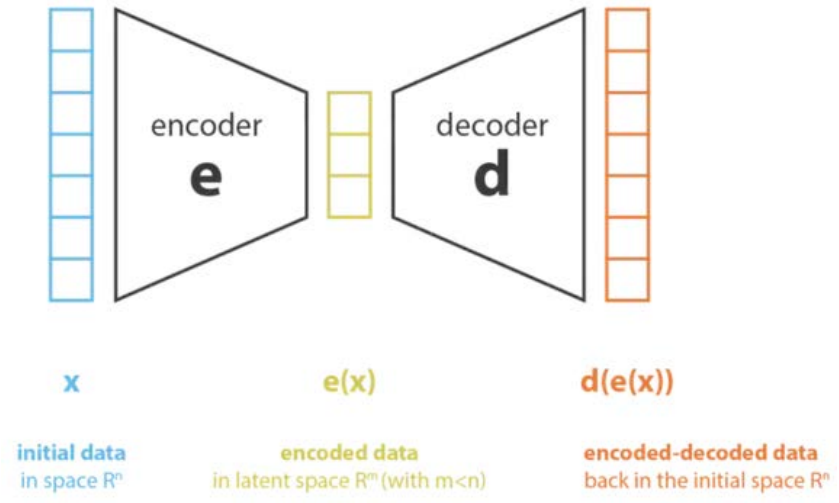


Figure 7: Visualization of how a VAE works.[28]

[25] https://doi.org/10.1016/j.ijepes.2014.03.005.

[26] Kleinebrahm, Max, Jacopo Torriti, Russell McKenna, Armin Ardone, and Wolf Fichtner. 'Using Neural Networks to Model Long-Term Dependencies in Occupancy Behavior'. *Energy and Buildings* 240 (June 2021): 110879. https://doi.org/10.1016/j.enbuild.2021.110879

[27] Wang, Chenguang, Simon H. Tindemans, and Peter Palensky. 'Generating Contextual Load Profiles Using a Conditional Variational Autoencoder'. In *2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, 1–6. Novi Sad, Serbia: IEEE, 2022. https://doi.org/10.1109/ISGT-Europe54678.2022.9960309

[28] https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Funded by the European Union

Funded by the European Union. 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.

Advantages:

- VAEs are robust to missing data and can be used to impute missing values in a dataset.
- For VAEs there is a clear and recognized way to evaluate the quality of the model.
- Compared to GANs, VAEs are easier to train.

Limitations:

- VAEs require a large amount of training data to accurately model the distribution of the data. This may be a challenge if the available training data is limited or if the load profiles are highly variable.
- While VAEs can handle sequential data to some extent, they are not specifically designed for this purpose and may not be as effective as other models, at modeling time series data.
- The decoder of the VAE does not work if it is used by itself. That means that a VAE always needs an input profile to generate an output profile.

1.2.3. Generative Adversial Network (GANs)

Since its introduction in 2014, **Generative Adversarial Networks (GANs)** have shown tremendous capabilities and potential to create realistic-looking images and generate synthetic data[29]. GAN belongs to the family of deep learning methods and consists of two neural networks, a discriminator (D) and a generator (G) (see Figure 8). The generator tries to produce data that is realistic enough to trick the discriminator, while the discriminator tries to correctly identify whether the generated data is real or fake. These two networks compete in the training process and reach an equilibrium when the generator is ready to generate synthetic samples that the discriminator cannot label as fake. Through this competition, the generator can learn and improve over time, generating realistic data.

GANs can be trained on historical load profiles to learn patterns and generate realistic load profiles. The generated profiles can be then used to simulate energy usage, predict energy consumption and demand, identify anomalies in real-worlds load profiles and augment real data to improve accuracy.

[29] I. Goodfellow et al., ‘‘Generative Adversarial Nets,” in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 26722680

Funded by the European Union

Funded by the European Union. 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.

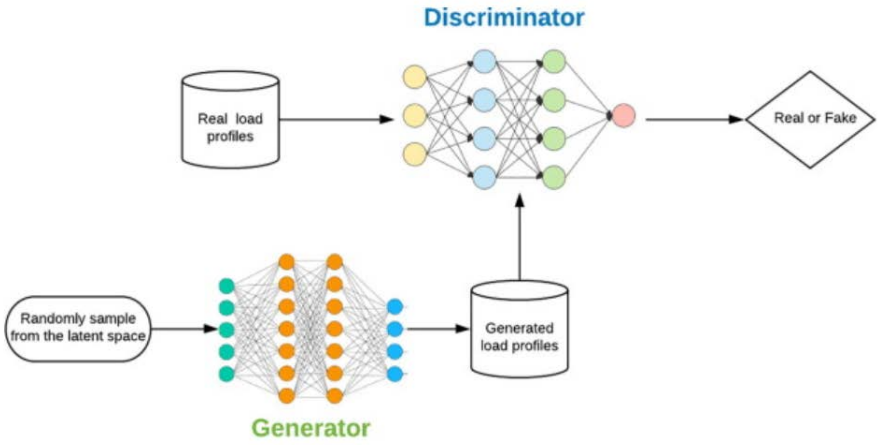


Figure 8: Visualization of the GAN principle30.

Advantages:

- GANs are known to be able to generate high-quality synthetic data that maintains the characteristic of the original data. This makes them well-suited for creating synthetic load profiles that keep the statistical and mathematical distribution of electricity usage patterns31,32,33,34.
- GANs preserve the privacy of the data and reduce the risk of original data information being compromised. Consequently, synthetic data can be stored and shared without privacy concerns35.
- GANs are flexible and adaptable techniques that can be applied to a wide range of data types and distributions. That makes it suitable to generate synthetic load profiles of different types of systems or environments.
- GANs can learn to generate synthetic data on the fly, which enables them to generate synthetic load profiles in real-time or near-real-time as needed.

[30] Wang, Zhe, and Tianzhen Hong. 'Generating Realistic Building Electrical Load Profiles through the Generative Adversarial Network (GAN)'. *Energy and Buildings* 224 (October 2020): 110299. https://doi.org/10.1016/j.enbuild.2020.110299;

[31] Wang, Zhe, and Tianzhen Hong. 'Generating Realistic Building Electrical Load Profiles through the Generative Adversarial Network (GAN)'. *Energy and Buildings* 224 (October 2020): 110299. https://doi.org/10.1016/j.enbuild.2020.110299;

[32] Li, Jianbin, Zhiqiang Chen, Long Cheng, and Xiufeng Liu. 'Energy Data Generation with Wasserstein Deep Convolutional Generative Adversarial Networks'. *Energy* 257 (October 2022): 124694. https://doi.org/10.1016/j.energy.2022.124694;

[33] Yilmaz, Bilgi, and Ralf Korn. 'Synthetic Demand Data Generation for Individual Electricity Consumers : Generative Adversarial Networks (GANs)'. *Energy and AI* 9 (August 2022): 100161. https://doi.org/10.1016/j.egyai.2022.100161;

[34] C. Zhang, S. R. Kuppannagari, R. Kannan and V. K. Prasanna, "Generative Adversarial Network for Synthetic Time Series Data Generation in Smart Grids," *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Aalborg, Denmark, 2018, pp. 1-6, doi: 10.1109/SmartGridComm.2018.8587464

[35] Venugopal, Rohit, Noman Shafqat, Ishwar Venugopal, Benjamin Mark John Tillbury, Harry Demetrios Stafford, and Aikaterini Bourazeri. 'Privacy Preserving Generative Adversarial Networks to Model Electronic Health Records'. *Neural Networks* 153 (Septem2022): 339–48.

Funded by the European Union

Funded by the European Union. 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.

Limitations:

- GANs are difficult to train and require domain knowledge in machine learning.
- GANs are sensitive to hyperparameters and may require careful tuning and optimization to produce high-quality results, which makes it difficult to achieve consistent and reliable performance.
- GANs are computationally intensive, which limits their applicability in scenarios where there are constraints on processing power or memory.

Due to its proven effectiveness to generate synthetic load profiles, GANs was selected as the tool to generate synthetic load profiles. A variety of GANs exist, but since it is also used to generate pictures or videos, not all of them are suitable to generate synthetic load profiles. The different GANs models differ in their architecture and loss function. Some common types of GANs include:

- Deep Convolutional GAN (DCGAN): This is a type of GAN that uses deep convolutional neural networks for both the generator and discriminator networks. It is commonly used for tasks such as image and video synthesis and super-resolution.
- Wasserstein GAN (WGAN): This is a type of GAN that uses the Wasserstein loss function to measure the distance between the generated data and the real data. It is known for its ability to produce high-quality and realistic outputs more consistently than a traditional GAN.
- CycleGAN: This is a type of GAN that is specifically designed for image-to-image translation tasks, such as converting a photograph of a horse into a painting of a horse. It uses a cycle-consistency loss function to ensure that the generated images are consistent with the input images.
- BigGAN: This is a type of GAN that uses a large-scale generator network with a hierarchical latent space to produce high-resolution and high-quality images. It is commonly used for tasks such as image synthesis and super-resolution.

GANs which are used to generate load profiles include:

- TimeGAN: This type of GAN is used to forecast future values in a time series. It is capable of generating accurate forecasts of multiple time-series variables, taking into account both short-term and long-term trends. TimeGAN is able to capture temporal correlations in the data, allowing for more accurate predictions than traditional methods. It can be used for a variety of tasks, such as energy demand.
- RCGAN stands for Recurrent Conditioned GAN and uses a recurrent neural network as the generator and the discriminator. Compared to standard GAN, RCGAN has the additional ability to generate data sequentially over time based on input conditions.
- Wasserstein GANs use the Wasserstein loss function to measure the distance between the generated data and the real data.
- Sig-Wasserstein GANs are a further variant of Wasserstein GANs using the Sinkhorn-Knopp algorithm to regularize the Wasserstein distance.

Yilmaz and Kron compare different GANs algorithms for individual electricity consumers[36], specifically: RCGAN, TimeGAN, CWGAN, and RCWGAN. TimeGAN also adopts a RNN as the generator and in addition, it has a loss function that uses a metric called the Earth Mover's Distance. It is specifically built to generate realistic time-series data. The Conditioned Wasserstein GAN (CWGAN) and the Recurrent Conditional Wasserstein GAN (RCWGAN) are both extensions of the Wasserstein GAN. The

[36] Yilmaz, Bilgi, and Ralf Korn. 'Synthetic Demand Data Generation for Individual Electricity Consumers : Generative Adversarial Networks (GANs)'. *Energy and AI* 9 (August 2022): 100161. https://doi.org/10.1016/j.egyai.2022.100161

Funded by the European Union

Funded by the European Union. 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.

CWGAN model uses additional information, known as "conditions," to guide the generation process. These conditions can be any type of information that is relevant to the data being generated, such as class labels for image data or temporal information for time-series data. The RCWGAN combines the capabilities of a RCGAN (relativistic training approach with recurrent networks) and a CWGAN (Wasserstein distance metric). In the study, they find that the CWGAN is better than the other three GANs in creating low electricity consumption profiles. The RCGAN performs best when generating high electricity consumption profiles. In all investigated GANs the statistics and distributional behaviors of the synthetic datasets are almost identical to the original data.

Another very promising GAN for generating synthetic load profiles was introduced by Lin et al[37]: DoppelGANger. This algorithm uses a "dual generator" architecture, which has two generator networks instead of just one. This allows the DoppelGANger model to learn from both "real" and "fake" data simultaneously, which can make it more effective at generating high-quality synthetic data. Additionally, the DoppelGANger model uses a technique called "variational inference" to improve the stability and performance of the GAN training process. In this approach, the Wasserstein loss function is also used. The model has been implemented by researchers involved in MODERATE (Python package)[38] and the Gretel-synthetics library[39]. After considering the pros and cons of each machine learning model for synthetic time series generation, we concluded that GANs is the most effective for the purpose. One of the main advantages of GANs is their ability to generate high-quality synthetic data. This can be particularly useful in cases where real-world data is scarce or hard to collect. Additionally, GANs can learn complex distributions and capture patterns in data that may not be easily captured by other models.

1.3. Generation of synthetic load profiles

The main objective is to explore and showcase the synthetic load profiles by utilizing a Generative Adversial Network (GAN) algorithm. To conduct an initial case study, we employed 395 electrical profiles with a time granularity of 1 hour over a period of 1.5 years. The dataset came with the timestamps and the loads of each profile, but without any metadata information, like type of building or geographic location.

In our study, we firstly used the DoppelGANger algorithm to generate synthetic load profiles. DoppelGANger is a GAN-based approach for generating synthetic time series data, which uses two neural networks to generate new data (see section 1.2.3 for detailed information concerning the internal processes of GAN's and particularly the doppelganger). DoppelGANger has been recently introduced by Lin et al from Carnegie Mellon University to overcome the challenge of the GANs approaches[40]. The traditional GAN framework has not been proven to be an effective method when working with time series data. Generative models should be able to capture the dynamic behavior, sequence, and pattern variation of time series data concerning various variables. Further, its effectiveness is determined by how it manages to maintain those relationships among the variables across the time while synthetic data being generated. Existing GANs have difficulty capturing long-term dependencies and complex multidimensional relationships, and address mode collapse.

[37] Lin, Zinan, Alankar Jain, Chen Wang, Giulia Fanti, and Vyas Sekar. 'Using GANs for Sharing Networked Time Series Data: Challenges, Initial Promise, and Open Questions'. In *Proceedings of the ACM Internet Measurement Conference*, 464–83, 2020. https://doi.org/10.1145/3419394.3423643.
[38] https://www.python.org/
[39] https://github.com/gretelai/gretel-synthetics
[40] Lin et al., 'Using GANs for Sharing Networked Time Series Data'.

Funded by the European Union

Funded by the European Union. 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.

DoppelGANger can capture the temporal dependencies between time series by using RNNs that generate batches instead of singletons. Metadata can influence the measurements of the time series, and DoppelGANger decouples it by associating time-series measurements with multi-dimensional metadata. The metadata feeds the time-series generator at each step and the architecture has an auxiliary discriminator to generate metadata. As a result, DoppelGANger achieves up to 43% better fidelity than baseline models and accurately captures the subtle correlations between data.

We used the implementation distributed by the company GretelAi[41], whose GANs architecture is already set up and is flexible to use. We first tested the algorithm using a reduced dataset by employing a meteorological season, **Winter** (1.12.2021 - 28.2.2022), for a total of 90 days, and a cluster of 62 load profiles. The sample length is of 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 10 000 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 performing. 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).

Unfortunately, the obtained results did not meet our expectations in terms of accuracy. DoppelGANger could not capture the intricacies of the real load profiles, probably because they are too dissimilar from each other (Figure 9).

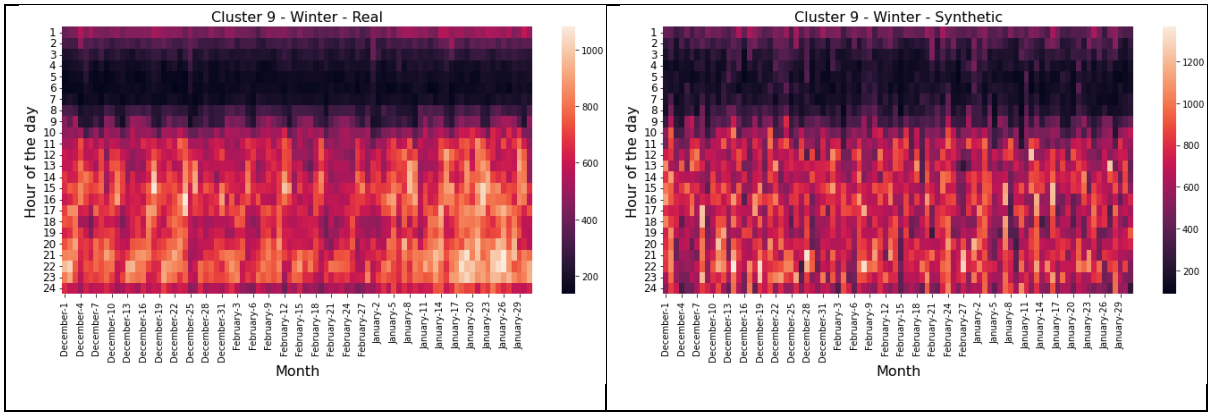


Figure 9: Agglomerative clustering of real data vs synthetic data

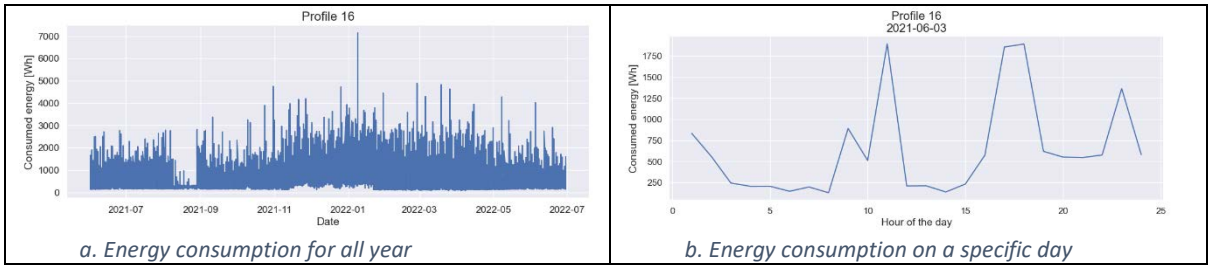
Our second approach was to consider a single profile (Figure 10). We first removed the incomplete days, and we trained using a similar DoppelGANger model as the one described above. The reliability and accuracy of DoppelGANger have been validated through the comparison with real load profiles. As shown in Figure 11 DoppleGANger proves to be highly efficient when it comes to generating synthetic load profiles for single days. Unfortunately, this is not the case for generating a profile on a

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

Funded by the European Union

Funded by the European Union. 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.

longer timeframe. Most probably, this is because the algorithm is tuned to capture temporal correlations and batches the samples rather than the singletons.



a. Energy consumption for all year *b. Energy consumption on a specific day*

Figure 10: energy consumption of a single load profile over one and half years (a) and for a single day (b)

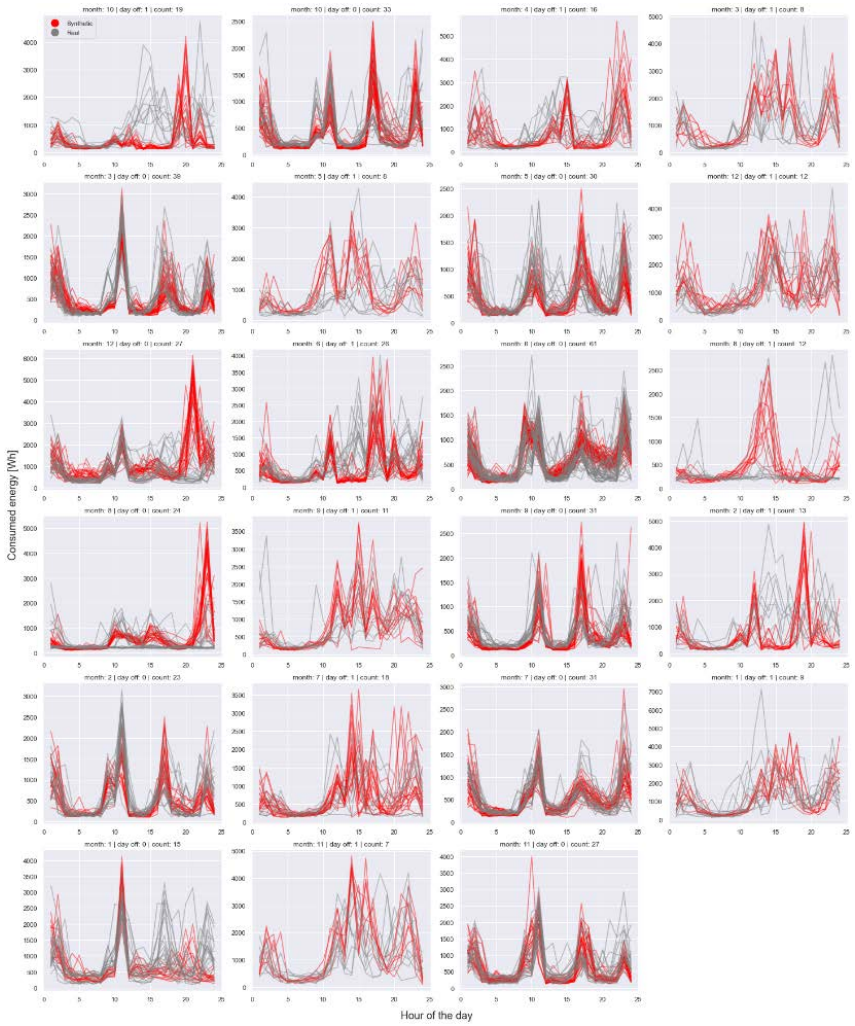


Figure 11: synthetic load profiles. The red profiles are the synthetic load profiles and the grey the real data

While DoppelGANger proves effective in generating data for a single day, we also assessed the performance of Conditional GANs (cGAN) for a more extensive time frame, such as a week or even a month.

The concept of **cGAN** was first introduced by Mehdi Mirza and Simon Osindero in 2014. The conditional GAN maximizes the performance of the generator and the discriminator by feeding them with class labels. Class labels are precise information of the dataset that guides the generator to

Funded by the European Union

Funded by the European Union. 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.

produce more specific data and the discriminator to better distinguish the generated data. In the context of electrical load profiles, the conditional information can include various factors like time of the day, day of the week, season, etc. cGAN has the same architecture of a traditional GAN: the generator takes both a random noise vector and the conditional information as input and generates synthetic data accordingly. The discriminator evaluates the authenticity of the generated samples, considering both the synthetic data and the conditional information. During training, the cGAN learns to generate synthetic electrical load profiles that not only resemble real profiles, but also satisfy the specified conditions. Once the cGAN is trained, it can be used to generate synthetic electrical load profiles by providing specific conditional information. For example, if we want load profiles for a specific time of day and season, we input that information into the conditioned generator, and it generates a synthetic load profile meeting those conditions.

For the test of cGAN, we reproduced a single day of 3 profiles from the same cluster. Data were normalized using MinMax scaler in the range [-1,1]. The neural network of the generator and the discriminator is architected on 6 layers with different numbers of nodes. The discriminator receives an initial vector of 24 values, while the generator has an input noise of 10 values in addition to the 2 labels. We used the Leaky ReLU activation function to prevent the problem of “dead neurons” (common in the ReLU), which makes the network inactive and stops learning. The leaky ReLU allows some information to pass through even when the input is negative and enhances robustness. We used the BCE as a loss function. Lastly, we saved the model every 500 epochs to better monitor the accuracy of the algorithm and turned the parameters accordingly. We noticed that the performance depends on the number of epochs and the size of the batches, the noise dimension that feeds the generator, and the number of nodes of the generator. The data produced through this method exhibited a striking resemblance to the authentic dataset. Specifically, the peak values in the profiles for each day closely matched those of the actual profiles. This occurred because the minimum and maximum values were preserved within the min-max scaler, and the synthetic data underwent scaling using identical scales for each day and profile. This poses a challenge when attempting to fully conceal personal data. Therefore, we adapted the model again. Normalisation of the data in the next step was done over all profiles the model is being trained with one minimum and one maximum value. The model then itself is supposed to learn the peak values of each day and the distribution of peaks within the profiles throughout the year, rather than just the shape of the profile.

In order to accomplish this, we used a clustering approach to sample the data into groups. This is necessary, because the original dataset contains diverse irregular profile, including energy consumption data from small offices, bars, and shops, as well as residential buildings with different uses. As there are no labels available and the profiles are only sorted by “contracted maximum power”, the clustering algorithm should group similar profiles together. This simplifies the training of a GAN, because the distribution of the data is not scattered as much and can be easier to approach. Additionally, we filterd out profiles that experience a very high peak demand. Profiles with peaks higher than the 95^{th} percentile are excluded from the training dataset. To validate how the dimension of the training profiles influences the results, we trained the model on the one hand with data split up into days which consist of 24 values (hours). On the other hand, we trained the model on the whole profiles, providing the model with the total number of days where each have 24 hours as well (e.g.: For one year one profile would correspond to a matrix of the dimension (365,24). In this format the GAN should learn the relation between days and between hours within each day over all provided profiles. However, it is questionable if the amount of provided profiles is large enough to train the model sufficiently as the differences between profiles is often high even within clusters. Also the GAN needs to be much more capable learning the total distribution at once instead of just single days which makes training more difficult. During the testing and validation phase, we will experiment with the

Funded by the European Union

Funded by the European Union. 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.

different settings explained above and also alter the structure of the GAN (different hidden layers of different sizes in both generator and discriminator).

Because GANs are notoriously difficult to train and they require large amounts of data for effective training, a VAE is also considered to generate synthetic load profiles. This solution might be more suitable for the MODERATE platform for costumers who do not want to synthesize excessive amounts of data. As a baseline the same architecture as explained in Wang et al.[42] is used. The data is pre-prepared as for the GAN. As a first start the profiles are conditioned with the same conditions as in the paper which are the *intensity*[43] of the load and the month. Within the testing and validation phase we are going to test the VAE set-up in more detail and compare it to the results of the GAN.

In conclusion GANs are a promising framework to generate synthetic data. The exact hyperparameters and model strucuter is still developed and will be finally be made publicly available at the end of the project. For smaller datasets the VAE could be a good alternative to the GAN.

[42] Wang, Chenguang, Simon H. Tindemans, and Peter Palensky. 'Generating Contextual Load Profiles Using a Conditional Variational Autoencoder'. In *2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, 1–6. Novi Sad, Serbia: IEEE, 2022. https://doi.org/10.1109/ISGT-Europe54678.2022.9960309

[43] The intensity of each profile is the maximum moving average value over 5 days.

Funded by the European Union

Funded by the European Union. 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.

2 Synthetic data generation for tabular data

In this chapter, we delve into the exploration of leveraging specific algorithms for the generation of synthetic data from tabular datasets. Tabular datasets, often deemed 'static,' encapsulate a spectrum of statistical information, including intricate correlations among diverse parameters. The objective of synthetic data generation, in this context, is to preserve these inherent statistical properties while generating entirely distinct datasets.

The synthesis of data, as explored in the subsequent sections, becomes a nuanced endeavor where the challenge lies in crafting new datasets that mirror the statistical nuances of the original tabular data. Our exploration particularly focuses on the application of genetic algorithms—a class of algorithms inspired by the process of natural selection—for the purpose of synthetic data generation.

In addition to elucidating the theoretical underpinnings of genetic algorithms in the context of synthetic data generation, we provide practical insights into their application through a detailed examination of their performance in two distinct open datasets. Through this comprehensive assessment, we aim to elucidate the potential and limitations of genetic algorithms in the realm of generating synthetic tabular data, contributing valuable insights to the broader discourse on data augmentation and diversification.

2.1. Overview of the models

Hereunder an analysis of possible machine learning models that can generate synthetic data from tabular data is presented.

Gaussian Copula is a mathematical model that is used to describe the dependence structure between multiple random variables. It is based on the concept of a copula, which is a function that describes the relationship between the marginal distributions of a set of random variables and their joint distribution. The Gaussian copula is particularly useful because it allows for the modeling of complex dependency structures in a relatively simple and tractable way. A Gaussian copula can be used to create synthetic data by first defining the marginal distributions for each of the variables that make up the dataset. Once the marginal distributions are defined, the copula function can be used to specify the dependence structure between the variables. This can be done by fitting a Gaussian copula to the original data, or by specifying the desired dependence structure directly. Therefore, a copula is a mathematical function that allows us to describe the joint distribution of multiple random variables by analyzing the dependencies between their marginal distributions.[44]

CTGAN: GAN (Generative Adversarial Network) is commonly used type of model for synthetic data. In the case of tabular data,. GANs offer greater flexibility in modelling distributions than their statistical counterparts. One of these is the CTGAN (Conditional Tabular Generative Adversarial Network). It is a generative model designed for generating synthetic tabular data. It is a variant of GAN that is specifically designed for handling tabular data and incorporates the knowledge of the underlying data distribution into the generator. CTGAN can be trained on real-world tabular datasets and can generate new samples that are similar to the original dataset, while preserving the underlying relationships between columns. CTGAN works by training a generator and a discriminator network in an adversarial manner. The generator network generates synthetic samples of the tabular data, while the discriminator network evaluates the generated samples and tries to distinguish them from the real data. The generator network is trained to generate samples that are similar to the real data, and the discriminator network is trained to correctly identify the generated samples as fake. The two networks

[44] https://sdv.dev/SDV/user_guides/single_table/gaussian_copula.html

Funded by the European Union

Funded by the European Union. 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.

are trained in an alternating fashion, and the training process continues until the generator produces synthetic samples that are indistinguishable from real data, according to the discriminator. In CTGAN, the generator and discriminator networks are conditioned on the statistical properties of the input data, such as the marginal distributions and dependencies between variables. This allows CTGAN to capture and preserve the relationships between variables in the generated data, making it well suited for generating synthetic data for tabular datasets. More information is available at [45] .

CopulaGAN is a variant of Generative Adversarial Networks that is specifically designed for generating synthetic multivariate data. Unlike traditional GANs, CopulaGAN models the dependency structure between variables in the generated data using copulas, a statistical tool for modeling dependencies between random variables. In CopulaGAN, the generator network generates synthetic samples of the multivariate data, while the discriminator network evaluates the generated samples and tries to distinguish them from the real data. The generator is trained to generate synthetic samples that are similar to the real data, and the discriminator is trained to correctly identify the generated samples as fake. The key difference between CopulaGAN and traditional GANs is that CopulaGAN incorporates the knowledge of the underlying dependency structure into the generator network. This allows CopulaGAN to generate synthetic data that not only resembles the real data in terms of its marginal distributions, but also preserves the dependencies between variables. CopulaGAN can be applied in various domains where multivariate data is generated, such as finance, insurance, and engineering, among others. It can be used to generate synthetic data for testing and validation, without compromising privacy or exposing sensitive information.

TVAE (Tabular Variational Autoencoder) is a generative model designed for generating synthetic tabular data. It is a type of Variational Autoencoder (VAE) that is specifically designed for handling tabular data, where the inputs are usually numerical or categorical variables. TVAE consists of two main components: an encoder network that maps the input data to a lower-dimensional latent representation, and a decoder network that maps the latent representation back to the original data space. The encoder and decoder networks are trained in an unsupervised manner, by minimizing a reconstruction loss that measures the difference between the input data and the reconstructed data. In addition to the reconstruction loss, TVAE also optimizes a regularization term that encourages the learned latent representation to have a desired distribution, such as a standard normal distribution. This regularization term helps the model capture the underlying patterns in the data and generate synthetic data that is similar to the real data.

A **Hierarchical Modeling Algorithm** is a type of statistical modeling approach that allows for modeling complex systems and data structures. In a hierarchical modeling approach, the data is decomposed into different levels or layers, with each layer representing a different level of abstraction or granularity. For example, in a hierarchical linear regression model, the data can be decomposed into different levels, such as individual observations, groups of observations, and higher-level aggregates. This allows for capturing the relationship between the different levels of the data and modeling the relationships between the variables in a more flexible and sophisticated manner.

In general, hierarchical modeling algorithms are useful in a wide range of applications, including population and sample estimation, causal inference, and clustering, among others. They are particularly well-suited for modeling complex data structures, where the relationships between the variables are non-linear or involve interactions between variables.

[45] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, Kalyan Veeramachaneni. Modeling Tabular data using Conditional GAN. NeurIPS, 2019

Funded by the European Union

Funded by the European Union. 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.

There are different types of hierarchical modeling algorithms, such as hierarchical linear regression, hierarchical Bayesian models, and hierarchical clustering algorithms, among others. Each type of algorithm has its own strengths and weaknesses and is suitable for different types of data and applications.

To this point a machine learning model was built for the dataset defined above using the algorithm *Hierarchical Modeling Algorithm with the Gaussian Copula Model.* It is an algorithm that allows to recursively walk through a relational dataset and apply tabular models across all the tables in a way that lets the models learn how all the fields from all the tables are related. More specifically, after the generation of multiple related tables from the original tabula dataset using metadata, the algorithm walks through all the tables in the dataset following the relationships specified by the metadata, learning each table using a **Gaussian Copula Model** and then augmenting the parent tables using the copula parameters before learning them. By doing this, each copula model was able to learn how the child table rows were related to their parent tables.

After creating synthetic data, the model is evaluated against the original data and a quality report is generated. This quality report evaluates how well the synthetic data captures mathematical data from the real dataset. The report is developed using the SDMetrics[46], which computes selected metrics (such as data fidelity and diversity) to measure data properties and summarizes the results. It is generated to evaluate if the synthetic data could be considered unique and not affect the privacy issue.

2.2. Datasets

Two distinct datasets have been examined to facilitate model generation from tabular data. The initial dataset[47] emanates from the LEIF organization and is openly available for the revitalization of their technology park. This comprehensive dataset encapsulates crucial information encompassing the energy consumption profiles, including thermal, electrical, and domestic hot water, across diverse building types such as schools, offices, and factories. The dataset provides a unique insight into energy usage patterns both before and after renovation activities. Complementing this, economic data pertaining to renovation costs, intervention types, unit energy costs per kWh, and outdoor temperatures from various weather stations have been incorporated. The latter is particularly useful for calculating heating and cooling degree days, offering a holistic perspective on the energy efficiency transformations introduced to the buildings.

The second dataset[48] focuses on Energy Performance Certificates (EPCs) publicly disclosed by the Lombardy region. Notably, this dataset represents the initial iteration following the antiquated energy classification standards for buildings. The inclusion of these datasets not only broadens the spectrum of available information but also allows for a nuanced exploration of energy-related patterns in distinct contexts, serving as a foundation for robust model development and analysis.

[47] https://data.mendeley.com/datasets/x6wyhmpj2v/2
[48] https://www.regione.lombardia.it/wps/portal/istituzionale/HP/DettaglioServizio/servizi-e-informazioni/Enti-e-Operatori/Ambiente-ed-energia/Energia/cened-certificazione-energetica-degli-edifici/cened-certificazione-energetica-degli-edifici

Funded by the European Union

Funded by the European Union. 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.

2.3. Application on EPC Dataset of Lombardia Region

In the specific case of EPCs, the goal is to achieve a more detailed prediction of the EPC value by requesting minimal building information from the user (such as window surface, building height, opaque and transparent components transmittance, surface, volume, location, construction year.) in addition to its geographical location. The work has been carried out using the opensource dataset from the Lombardy region in Italy (the first version). Following data cleaning, the dataset resulted in 255 666 EPCs that could be utilized. Subsequently, a neural network based on the MultiLayer Perceptron regression algorithm was implemented. With a current accuracy of 90%, it can identify the energy class of the building and, consequently, its energy consumption.

The main challenges arise when evaluating highly efficient buildings classified as A+ (old classification) since the available data are not sufficient to train the network. In these buildings, the model produces the largest errors, especially because the consumption differences between higher energy classes (A and A+) are not as significant as in other classes.

In order to enhance the model's performance, more data is required. In this case, due to the lack of available data, the decision was made to utilize generative algorithms to create new data, especially for high-performance buildings. The initial approach involved employing the aforementioned algorithms, the validation of which is still ongoing. The results will subsequently be provided to the Neural MLPRegressor model for the aforementioned purposes.

2.4. Application on LEIF Dataset

A machine learning model was built for the dataset of LEIF using the algorithm *Hierarchical Modeling Algorithm* which is an algorithm that allows to recursively walk through a relational dataset and apply tabular models across all the tables in a way that lets the models learn how all the fields from all the tables are related. After creating synthetic data, the model is evaluated against the original data.

A specific quality report is generated. It evaluates how well the synthetic data captures mathematical data from the real dataset. The report is developed using the SDMetrics[49] and it is also known as synthetic data fidelity. The report runs select metrics to measure data properties and summarizes the results. It is generated to evaluate if the synthetic data could be considered unique and not affect the privacy issue. For more information refers to : https://docs.sdv.dev/sdmetrics/

Hereunder the result of the analysis carried out for the LEIF dataset.

2.4.1. SYNTHESIS

The synthesis metric measures whether each row in the synthetic data is novel, or whether it exactly matches an original row in the real data.[50]

This metric looks for matching rows between the real and synthetic dataset. In order to be considered a match, all the individual values in the real row must match the synthetic row. The exact matching criteria is based on the type of data. More information regarding the calculation used is available here:

https://docs.sdv.dev/sdmetrics/metrics/metrics-glossary/newrowsynthesis

[49] Synthetic Data Metrics. Version 0.8.0. DataCebo, Inc. Oct. 2022. URL: https://docs.sdv.dev/sdmetrics/

[50] https://docs.sdv.dev/sdmetrics/metrics/metrics-glossary/newrowsynthesis

Funded by the European Union

Funded by the European Union. 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.

The metric provides a score between:

- **0.0** (worst): All the rows in the synthetic data are copies of the row in the real data
- **1.0** (best): The rows in the synthetic data are all new. Figure 12 shows that there are no matches with the real data for our case.

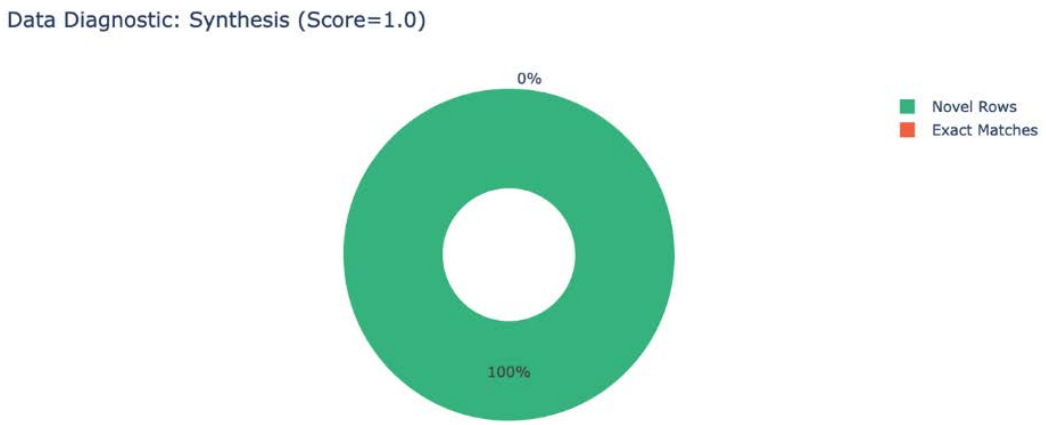


Figure 12 Data Diagnostic Synthesis

2.4.2. DIAGNOSTIC REPORT

The Table generated in the Quality Report (Table 1) evaluates how well your synthetic data captures mathematical properties in the real data. *(src: https://docs.sdv.dev/sdmetrics/reports/quality-report/single-table-quality-report).*

Table 1 Diagnostic Report for LEIF model

Show 10 entries Search:

	SUCCESS	WARNING	DANGER
1	Over 90% of the synthetic rows are not copies of the real data	The synthetic data is missing more than 10% of the numerical ranges present in the real data	
2	The synthetic data follows over 90% of the min/max boundaries set by the real data	The synthetic data is missing more than 10% of the categories present in the real data	

Showing 1 to 2 of 2 entries Previous 1 Next

The following graph (Figure 13) shows the 'coverage' of each parameter. This metric measures whether a synthetic column covers all the possible categories that are present in a real column.[51]

[51] ttps://docs.sdv.dev/sdmetrics/metrics/metrics-glossary/categorycoverage#does-high-coverage-that-mean-my-synthetic-data-is-similar-to-the-real-data

Funded by the European Union

Funded by the European Union. 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.

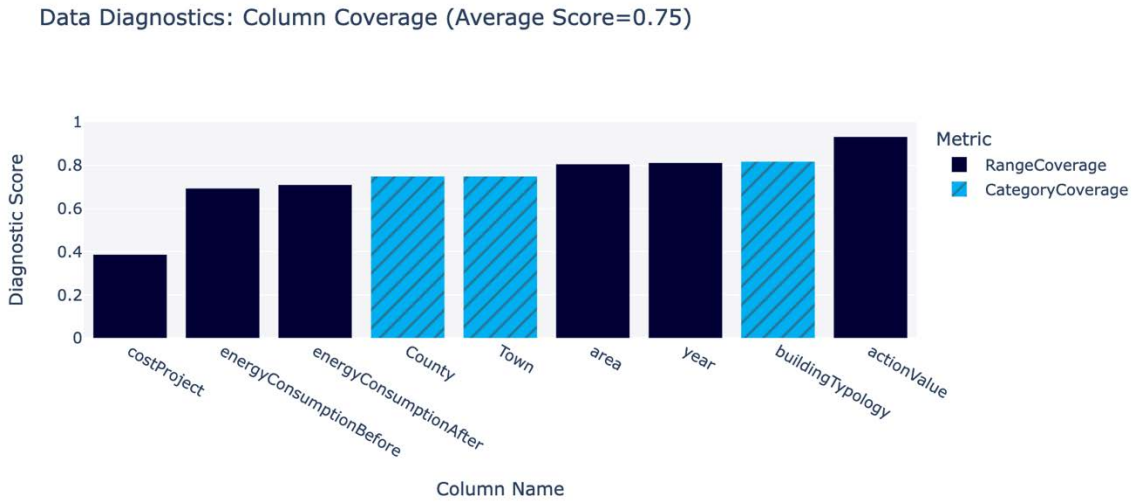


Figure 13 Colum Coverage

2.4.3. BOUNDARY ADHERANCE

This metrics measures whether a synthetic column respects the minimum and maximum values of the real column. It returns the percentage of synthetic rows that adhere to the real boundaries.[52] The metric is applied on each parameter and provide a graph (Figure 14) and a score (Table 2). The score could be

- **0.0** (worst): No value in the synthetic data is in between the min and max value of the real data
- **1.0** (best): All values in the synthetic data respect the min/max boundaries of the real data

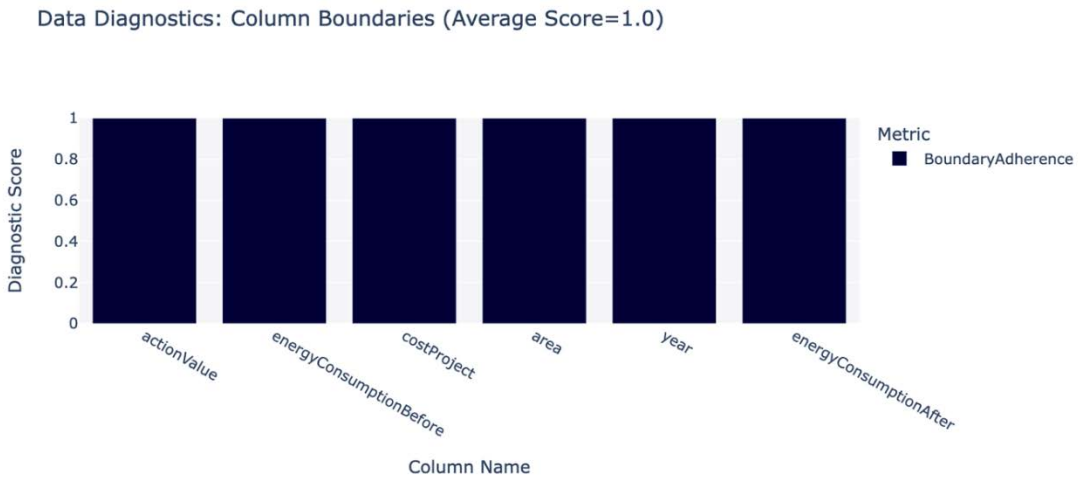


Figure 14 Data Diagnostics: Column Boundaries

[52] https://docs.sdv.dev/sdmetrics/metrics/metrics-glossary/boundaryadherence

Funded by the European Union

Funded by the European Union. 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.

Table 2 Boundary adherence for LEIF dataset

Show 10 entries Search:

	parameters	value
1	buildingTypology	1
2	bui_id	1
3	actionValue	1
4	County	1
5	Town	1
6	energyConsumptionBefore	1
7	costProject	1
8	area	1
9	year	1
10	energyConsumptionAfter	1

Showing 1 to 10 of 10 entries Previous 1 Next

The following graphs show the distribution of the synthetic data (generated by the model) compared to the original data for each variable (from Figure 15 to Figure 23) of the dataset (Country, building_typology, town, energy consumption before building refurbishment, energy consumption after building refurbishment,cost of sigle refurbishment project, building area and refurbishment action applied. The latter is described in the following.

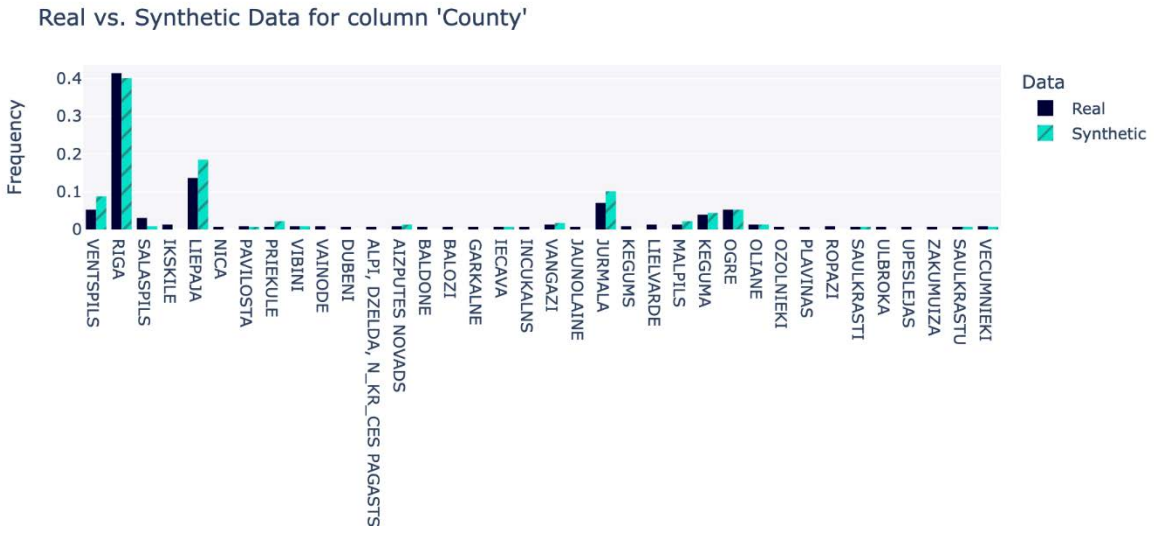


Figure 15 Distribution of real and syntethic data for column Country

Funded by the European Union

Funded by the European Union. 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.

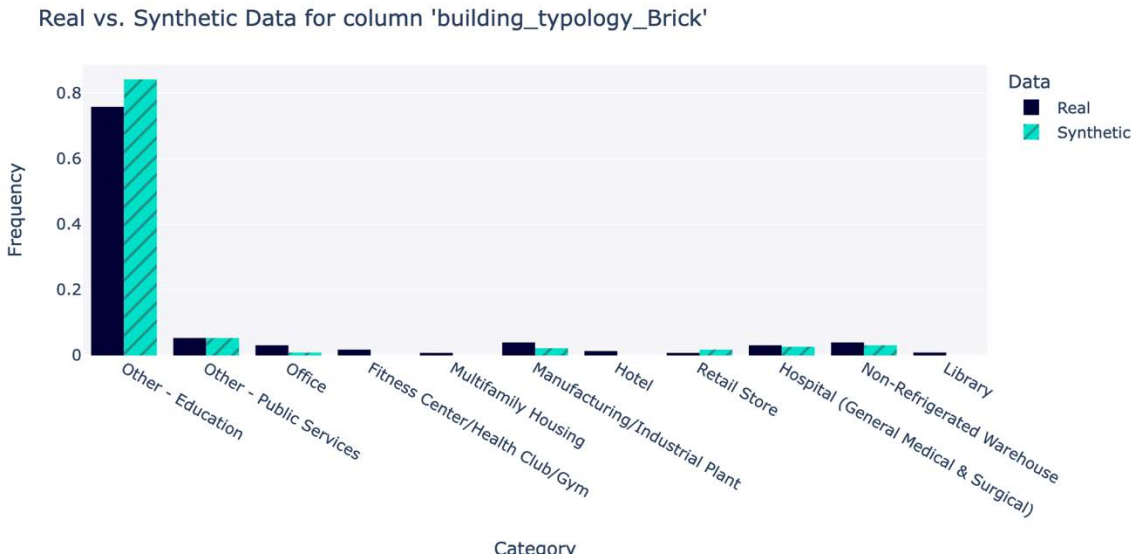


Figure 16 Distribution of real and syntethic data for column building typology by Brick schema

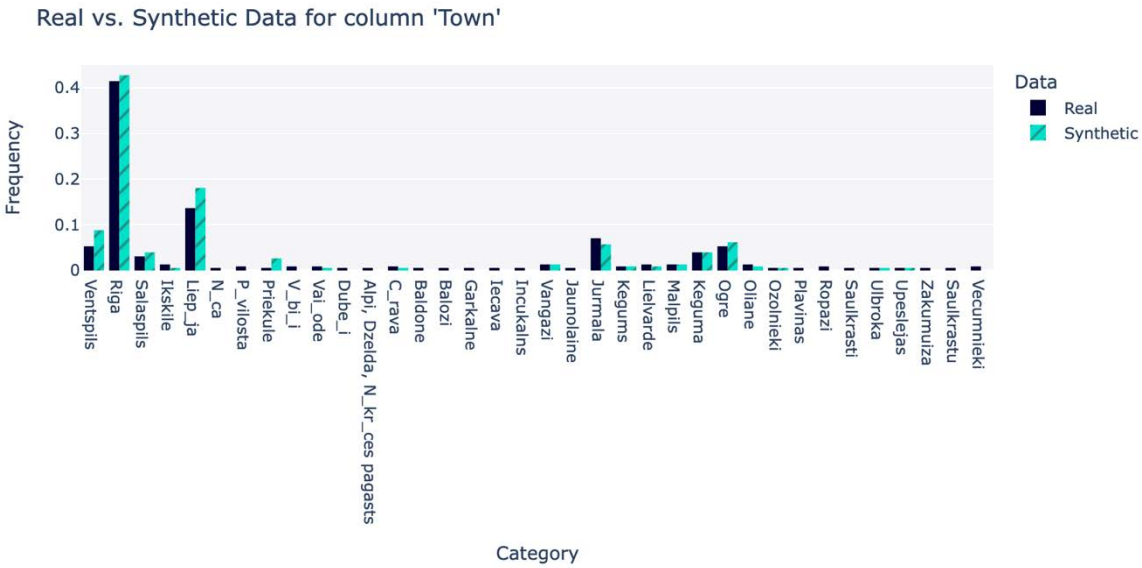


Figure 17 Distribution of real vs synthetic data for "town" propriety of LEIF dataset

Funded by the European Union

Funded by the European Union. 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.

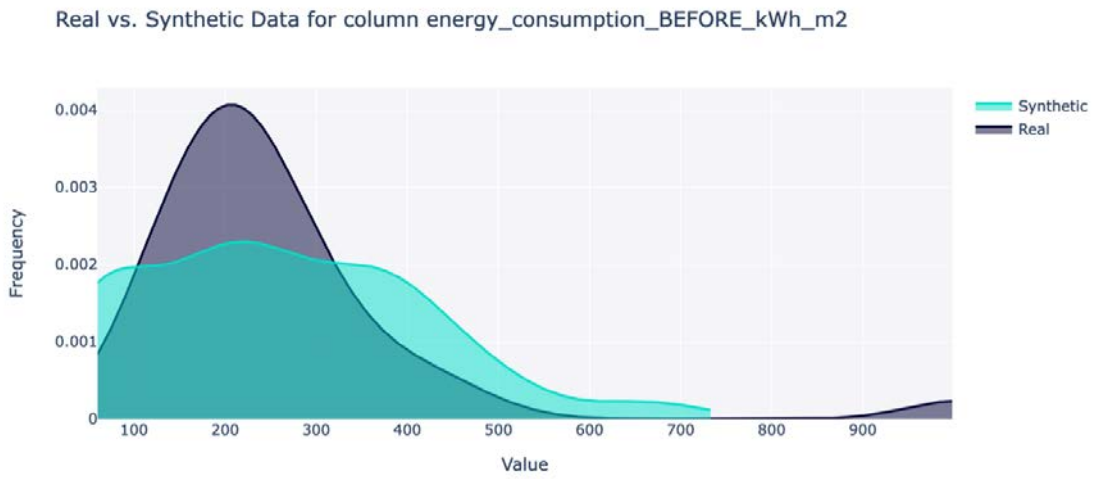


Figure 18 Distribution of real vs synthetic data for "energy consumption before refurbishment" propriety of LEIF dataset

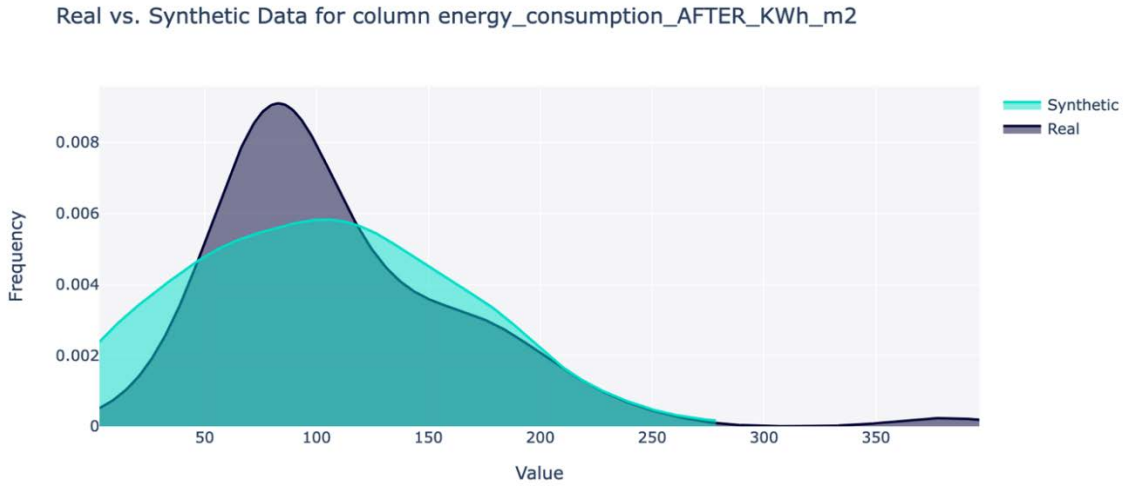


Figure 19 Distribution of real vs synthetic data for "energy consumption after refurbishment " propriety of LEIF dataset

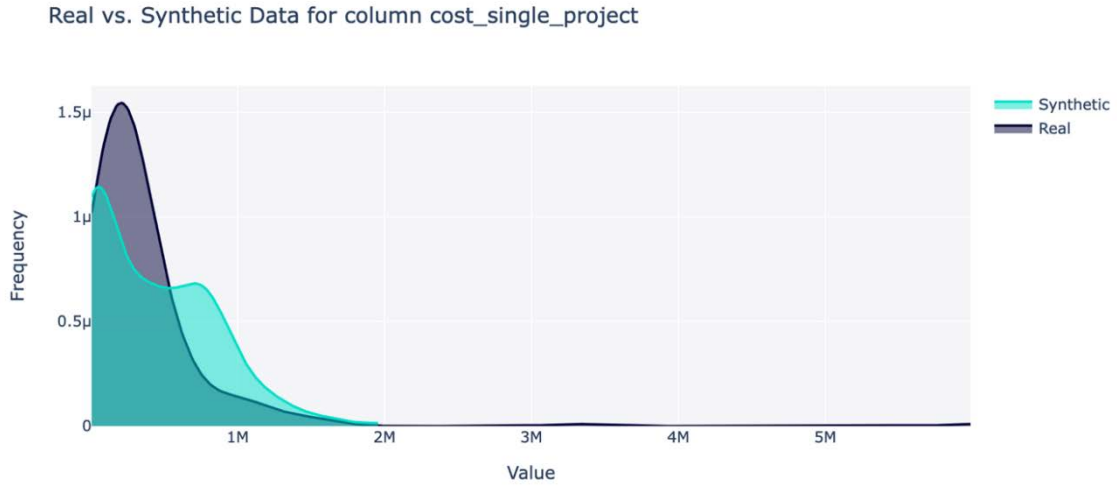


Figure 20 Distribution of real vs synthetic data for "cost single project" propriety of LEIF dataset

Funded by the European Union

Funded by the European Union. 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.

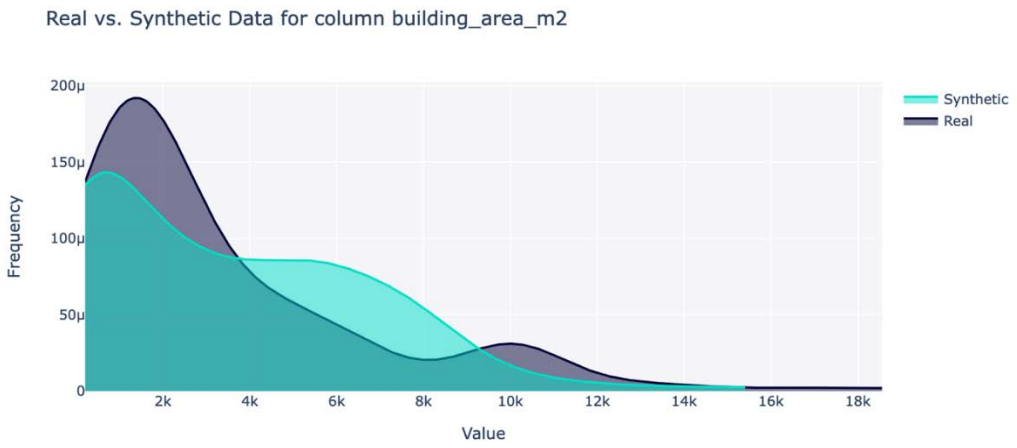


Figure 21 Distribution of real vs synthetic data for "building area" propriety of LEIF dataset

Possible renovation actions

Type	Action Description	N°
ACTION 1	renovation of the building's enclosing structures	0
ACTION 2	renovation of the building's enclosing structures, ventilation system renovation	1.01
ACTION 3	renovation of the building's enclosing structures, energy efficient lighting	1.02
ACTION 4	renovation of the building's enclosing structures, heat supply renovation	1.03
ACTION 5	renovation of the building's enclosing structures, energy efficient lighting, ventilation system renovation	2.01
ACTION 6	renovation of the building's enclosing structures, ventilation system renovation, heat supply renovation	2.02
ACTION 7	renovation of the building's enclosing structures, energy efficient lighting, heat supply renovation	2.03
ACTION 8	renovation of the building's enclosing structures, energy efficient lighting, heat supply renovation	2.04
ACTION 9	renovation of the building's enclosing structures, energy efficient lighting, heat supply renovation, ventilation system renovation	3.01
ACTION 10	renovation of the building's enclosing structures, energy efficient lighting, heat supply renovation, technological equipment	3.02

Figure 22 List of refurbishment actions

Funded by the European Union

Funded by the European Union. 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.

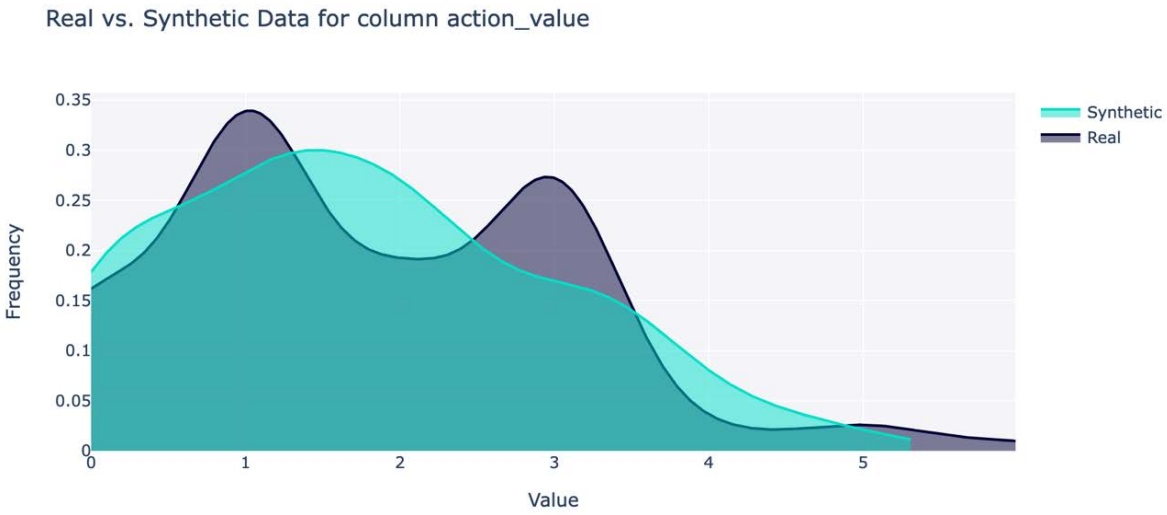


Figure 23 Distribution of real vs synthetic data for "action value" propriety of LEIF dataset

As can be seen from the figures above, the model is able to generate data that closely resembles real data with similar statistical characteristics, being original in itself. This approach is currently under testing for tabular data from the Energy Performance Certificates (EPCs) as well.

2.5. Results

The procedure descrived in the previous chapter illustrates the validation of the synthetic data generation model. It is apparent that, in certain instances, the boundary values of a distribution may not be replicated perfectly, aligning with the intentional nature of synthesis. The objective is not to produce perfectly identical data, as doing so could give rise to privacy concerns. Hence, striking a balance between creating new synthetic and mimicking real data is a pivotal challenge. More specifically, the goal is to pinpoint and replicate the statistical characteristics of the dataset while accommodating variations in the data distribution. This delicate equilibrium ensures that the synthetic data maintains its utility in various applications without compromising the privacy and integrity of the original dataset.

Funded by the European Union

Funded by the European Union. 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.

3 Machine Learning Techniques for Building Stock Characterization

Chapter 3 to 5 describe the machine learning techniques and methodologies for building stock characterization.

The characterization of building stock involves the formulation of representative buildings (or Archetypes). Such characterization is driven by the availability of urban building stock data that might include geometrical parameters, building characteristics, operational energy use, street view imagery and building footprints. Previous literature in the built environment ecosystem uses several data-driven machine learning techniques to achieve this characterization though a process termed as classification. Within the machine learning and artificial intelligence domain, there exists another process workflow, called, clustering that has also been widely used for characterizing the urban building stock. Classification is a supervised learning approach where a specific label is provided to the machine to classify new observations. Clustering is an unsupervised learning approach where grouping is done on similarities basis. The major difference between classification and clustering is that classification includes the labelling of items according to their membership in pre-defined groups.

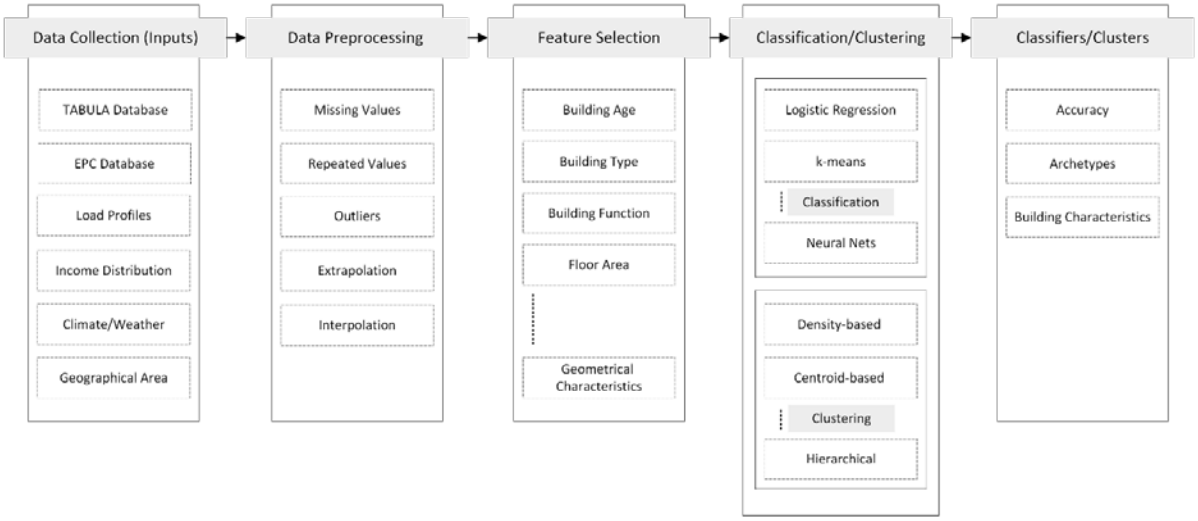


Figure 24 The devised workflow to identify classifiers/clusters from building stock datasets using different data-drive machine learning techniques.

The characterization process differs based on the type and the availability of data. Furthermore, the implementation of classification or clustering techniques depends on the parameters/features present in the dataset. It is worthwhile to note that the existing ML techniques are eventually compared in their ability to classify/cluster the available data. There are various datasets that act as inputs to the ML models for synthetic data generation. This task underlines techniques to deal with data scarcity. Moreover, the task elaborates on formulating relations between different databases.
The devised workflow evaluates the different publicly available datasets under the data collection step. Commonly used datasets in this task include the TABULA database, available EPC national/sectoral/regional datasets, end-use time varying load profiles, district/street level aggregated energy consumption, national income distribution, climate/weather datasets and geographical area information. These datasets include different levels of information; each dataset enriches the building stock model from a top-down or a bottom-up perspective. For instance, in the event of data scarcity, the workflow initiates the building stock with the TABULA database for one specific region. When the regional EPC data becomes available, the existing TABULA characterization

Funded by the European Union

Funded by the European Union. 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.

is further reformulated using building features/parameters available in these dataset. This characterization is further enriched if there are any existing end-use time series demand profiles. Other national level datasets such as income distribution, climate data and geographical area layer enriches the existing characterization with district wide parameters.

The data pre-processing step improves these data layers and treats the data for any missing or repeated values, removes outliers and performs extrapolation/interpolation techniques for structuring the original dataset. Since such datasets involves numerous parameters that might not be equally influential, the feature selection process filters building features/parameters to list out the most influential ones. This is either done through expert opinion or feature extraction techniques. The classification/clustering step then implements various data-driven machine learning techniques to actively identify classifiers or clusters. This step defines inputs and outputs depending on the availability of data. For instance, classifiers are mainly used when detailed EPC data is not available. The clusters are formulated when detailed EPC data or load profiles are available. Each classifier/cluster is associated with an accuracy depending on the various ML techniques. Each classifier/cluster represent an archetype, which has fixed building characteristics as derived from the datasets.

Commonly used classification and clustering techniques are elaborated in the following sections. These techniques are also used for building energy performance data generation and load disaggregation analysis.

3.1. Synthetic Urban Building Energy Performance Data Generation

Synthetic building energy performance data generation goes one step beyond the classification/clustering process workflow and deploys a data-driven physics-based technique. The first four steps in this workflow, namely data collection, data pre-processing, feature selection and building clusters are similarly formulated for the building clustering workflow. This workflow mainly uses the generated archetypes as inputs to create a physics-based white-box model. Such models are computationally intensive when considering the entire district wide building stock. However, the use of typical representations or archetypes reduces the computational burden by a significant amount. Whenever required, the original data stock could be recreated using a parametric procedure to vary the influential building features/parameters.

The simulation process workflow (Figure 25) initiates the building stock model using the regional EPC database. A fact check is done to ensure that the EPC data enriches the already available information in the TABULA database. The data pre-processing and feature selection steps improve the formulation of building clusters, which act as the foundation for generating building energy data. The data-driven physic-based model use the parameters defined in these clusters as inputs to the physics driven model. This task uses the EnergyPlus software to calculate the energy demand associated with each archetype. EnergyPlus is a free, open-source and cross platform to run whole building energy simulations and reads inputs and writes output to text files. EnergyPlus implements detailed building physics for air, moisture, and heat transfer including treating radiative and convective heat-transfer separately to support modeling of radiant systems and calculation of thermal comfort metrics; calculates lighting, shading, and visual comfort metrics; supports flexible component-level configuration of HVAC, plant, and refrigeration systems; includes a large set of HVAC and plant component models; simulates sub-hourly time steps to handle fast system dynamics and control strategies; and has a programmable external interface for modeling control sequences and interfacing with other analyses. The building parameters are fed into EnergyPlus Input Data File (IDF) to create a building energy simulation model. The first run of this model is considered as the generated baseline of the building energy data. The simulation runs following the baseline generate additional data of the entire building stock associated

Funded by the European Union

Funded by the European Union. 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.

with an individual archetype. These simulation runs are generated using parametric techniques that randomize the parameter extraction from a probability distribution function (PDF) curve. The EPC dataset is used to generate these PDFs for each of the influential parameters. This task demonstrates the entire workflow to generate synthetic load profiles, calculates energy savings associated with efficiency improvements and defines aggregated demands for building in a particular cluster.

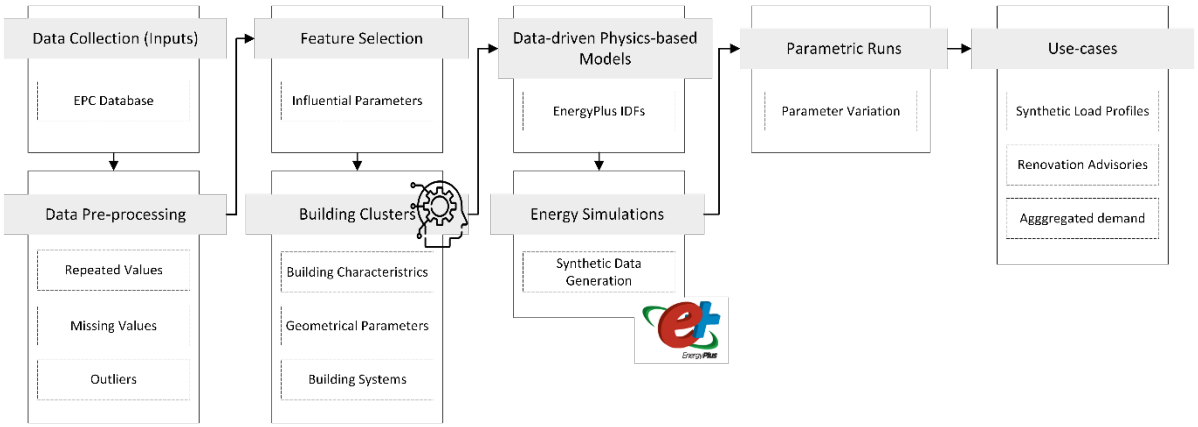


Figure 25 The simulation process workflow that uses a combination of data-driven and physics-based models to generate synthetic building energy performance data.

3.2. Building stock data disaggregation

Building stock data disaggregation refers to the process of downscaling aggregated or summarized high level data into more detailed, granular information to lower level. In this task, 3 levels/types of disaggregation are being explored and investigated, with different purposes:

- Disaggregation of national or regional static data to enrich archetypes
- Disaggregation of local level GIS based data to individual building level
- Disaggregation of building level energy consumption (load profiles) to end service level

Disaggregated national or regional static data (e.g. construction year, building size, construction type, insulation level, energy usage patterns, or specific technologies in use) often serves as inputs to enrich, validate or calibrate the characteristics of existing archetypes. These refined archetypes are then utilized in building stock models to enhance the accuracy and granularity of the energy analysis. Various national level datasets are available and could be used for this purpose, such as EU BSO[53] and JRC-IDEES dataset [54].

When it comes to district or street level datasets, such as GIS based aggregated energy consumption for a specific district, downscaling measured gas or electricity consumption from aggregated zip code levels to individual buildings with linear regression models is a commonly used approach. The process typically involves developing regression models based on relevant features to estimate energy consumption at a more granular level. Primarily, a correlation analysis can be conducted to identify relationships between street-level energy consumption and building characteristics. This may involve statistical analysis to determine which building features are strongly correlated with energy usage. Possible ones include building area, heated volume, heating system type, insulation level, and number of occupants. After the most influential features being identified, the next step is to use them to create

[53] https://building-stock-observatory.energy.ec.europa.eu/database/
[54] https://data.jrc.ec.europa.eu/dataset/jrc-10110-10001

Funded by the European Union

Funded by the European Union. 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.

(multiple) linear regression models. This method will be tested in the next phase, using open source datasets, the street level gas and electricity consumption data in Flanders[55]. The street level annual consumption data is available per energy carrier (electricity/gas), per main municipality at street level. With more detailed GIS based building characteristics data is available, physics based whitebox models can be used to simulate building energy consumption patterns individually, then the actual energy consumption can be disaggregated to building level. This approach might be computational expensive. Smart energy meter data of individual buildings can be used as validation.

Next, disaggregation of the building stock data adds another layer of information to the existing dataset in the form of time-series load profiles that are either measured or synthetically generated using physics-based models. Disaggregation often refers to the decomposition of end-use loads, namely electricity and heat demand. For instance, building heat demand could be further disaggregated into space heat and domestic hot water demands. Disaggregation methodologies often use event detection techniques when highly resolute data is available or nonevent detection techniques using steady state power levels. At the urban building stock level, this normally entails the use of clustering techniques to identify load shaped clusters, e.g., building occupancy types (restaurant, retail, warehouse etc.).

The disaggregated data generation workflow (Figure 26) initiates the building stock models using the building load profiles. These profiles might include electricity consumption, gas consumption or internal temperature. Such load profiles are further refined to remove duplicates, fill in any missing values and manage outliers. The clustering process then uses these profiles to identify similar load shaped clusters based on rooftop solar, electric vehicle and building function or occupancy types. These clusters are further enriched with geographical area information, local climate data, income distribution and aggregation consumption data. Disaggregation of the demand could be either temperature dependent or temperature independent. For instance, the workflow uses the gas consumption data and associates these profiles with the outside dry bulb temperature to generate space heating (SH) and domestic hot water (DHW) demand independently. This identifies a critical point during the day when the outside temperature is above the setpoint, and the heating does not operate in the building. This essentially corresponds that the gas consumption is solely due to the use of domestic hot water within that period. The generated decomposition (SH and DHW) data further refines the building stock models with additional end-use information. Alongside, these act as inputs to further refine building parameters using building measurements (defined as calibration).

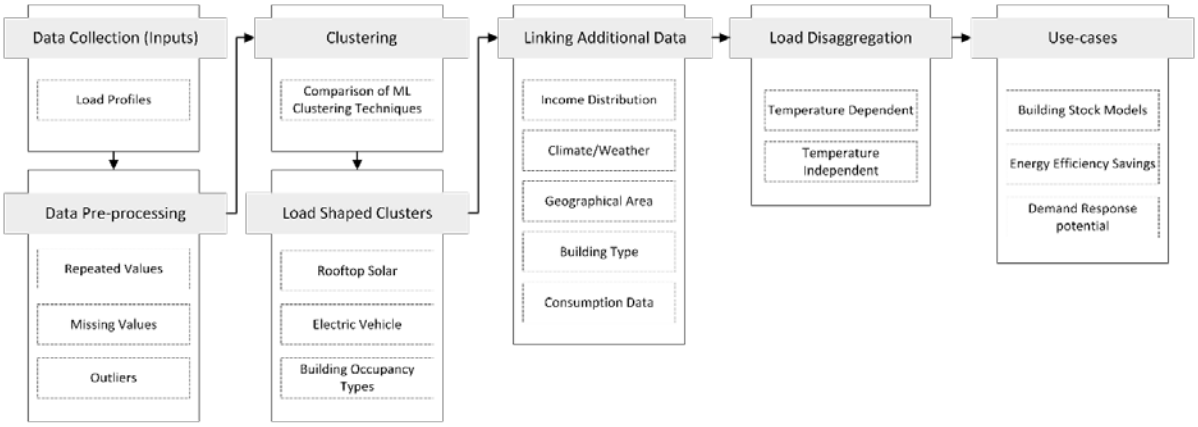


Figure 26 The disaggregation process workflow to generate decomposed end-use demand data.

[55] https://opendata.fluvius.be/explore/dataset/1_03-verbruiksgegevens-op-straatniveau/information/

Funded by the European Union

Funded by the European Union. 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.

3.3. HVAC identification

One important feature associated to each building is the installed heating, cooling, and air conditioning (HVAC) system. HVAC directly impacts all the energy related analysis in the building sector. Indoor air quality, district energy vector analysis, CO_2 emissions reduction plans, building retrofit plans are among the most important energy related topics that need HVAC identification. Moreover, the analysis in the district level requires a geospatial allocation due to the need for specifying the location of the energy carrier use. This section explains ML techniques that can help in identification and geospatial allocation of HVACs.

HVAC identification can be conducted using classification techniques. Classification techniques are especially efficient algorithms for predicting categorical outcomes. Trained using building related parameters, the classification model can predict the type of the installed HVAC at the frost step. In a later stage, the efficiency of the HVAC and other features if needed, can be predicted by the model. The procedure of HVAC identification also depends on the available input data characteristics. Input data is characterized using three indicators 1) granularity, 2) quality, and 3) frequency of the available data. If the accessible input data are on district level, as a preprocessing step, the disaggregation is applied on the input data and the individual building level dataset is provided. In a next step, data enhancement methods are applied to identify HVAC of the building. As depicted in Figure 27, to identify the HVAC a variety of data enhancement methods are applied to different datasets depending on the characteristics of the original dataset.

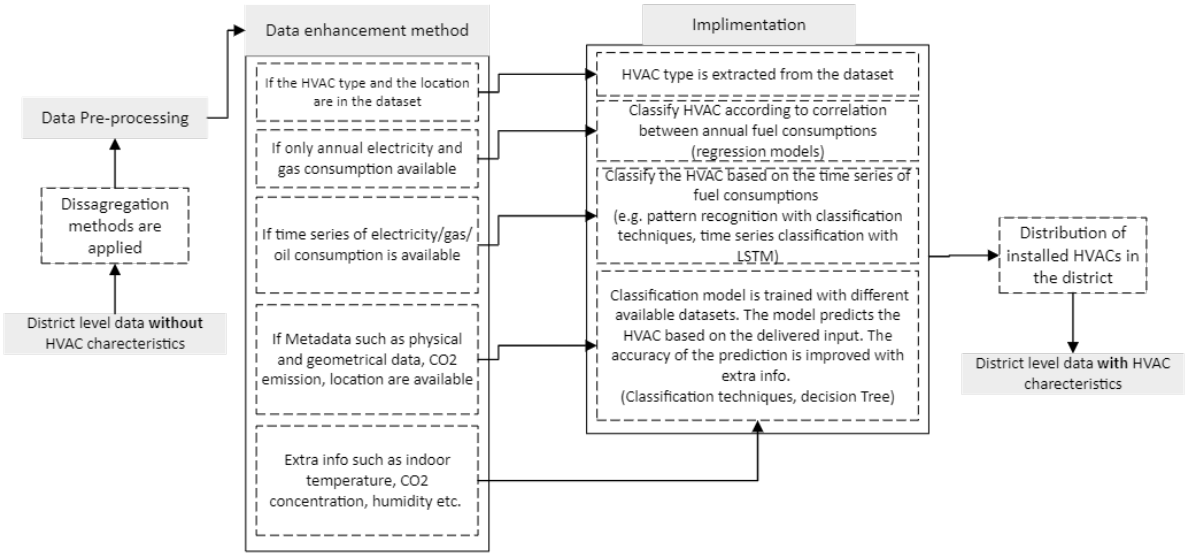


Figure 27 High level flowchart of proposed multistep data enhancement methods for HVAC geospatial identification

Figure 28 shows an example of hourly time series of measured electricity consumption for heat pump, appliances, and their summation as total electricity consumption.

Funded by the European Union

Funded by the European Union. 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.

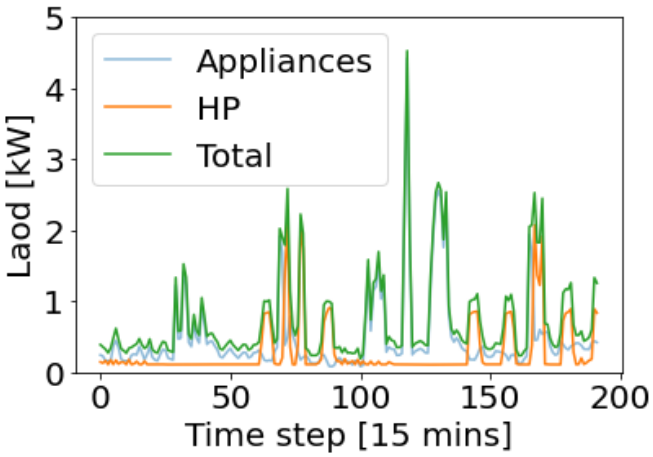


Figure 28 Example of electricity consumption (from the dataset used in this study) for a building with and without heat pump.

The difference between electricity consumption time series with and without heat pump are used in training ML models. The model will later be used to predict if HP is installed in a building associated to a given time series. The same procedure can be adopted to train ML models that can predict installed PV-panels, gas boiler (according to gas consumption), building appliances. The procedure starts with exploratory data analysis (EDA) to identify features to use for training the model. This step is specific to supervised learning methods. Figure 29 exemplifies a comparison between load duration curves of the electricity consumption for two case studies with different HVAC installed. The comparison depicts the differences between peak demands, low power zone characteristics, and relation between total demand and peak power in an abstract manner. The next step is to quantify the difference for choosing appropriate features.

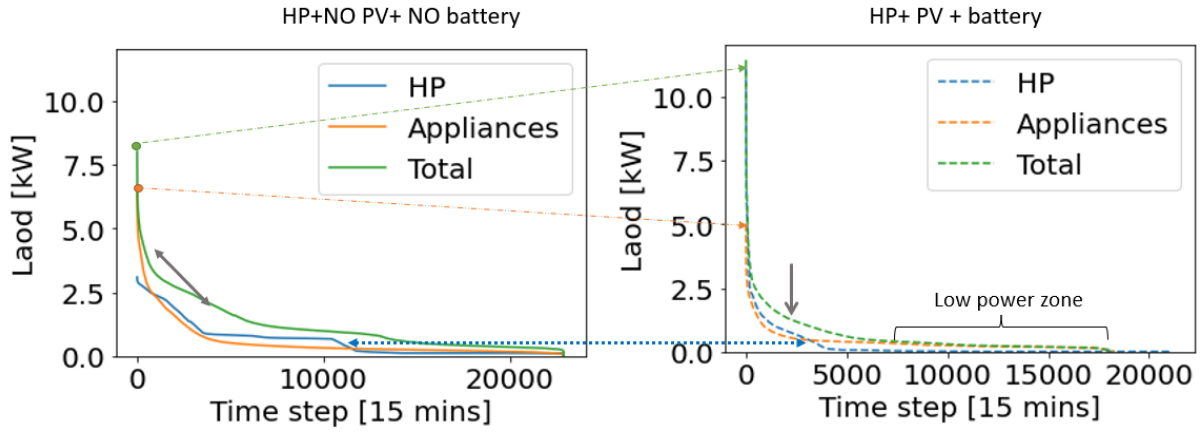


Figure 29 Example of a comparison between load duration curves (with 15 mins time step) of the electricity consumption for two case studies with different HVAC installed.

Common features of time series for ML models are typically relations between the values in different periods. In supervised learning, the period can be derived from physical definitions such as hours, days, and months. These definitions relate to the cyclic activities of occupants in a building. A data set of 150 time series with known installed HVAC are sued in this exercise. We define 13 combined features which are 1 feature as annual peak divided by annual average power and 12 features as monthly peak power divided by annual average power (corresponding to 12 months of a year also named hereafter ind_M_1 to ind_M_12). Note that the first 4 features don't have value in this exercise

Funded by the European Union

Funded by the European Union. 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.

because measurements were not available for the first 4 months of the year in the case studies. However, we kept the features in the training and test sets to investigate whether the method is robust to missing data. Features as explained, ind_M_1 to ind_M_12 is calculated for each time series and HVAC as labels are assigned as was available in the description of buildings associated to each time series. This generates a new dataset that is used for training the model. 70% percent of the dataset, as a conventional approach, is picked for the training step, and the rest are used for the testing the model. The distribution of the labels is shown in Figure 30.

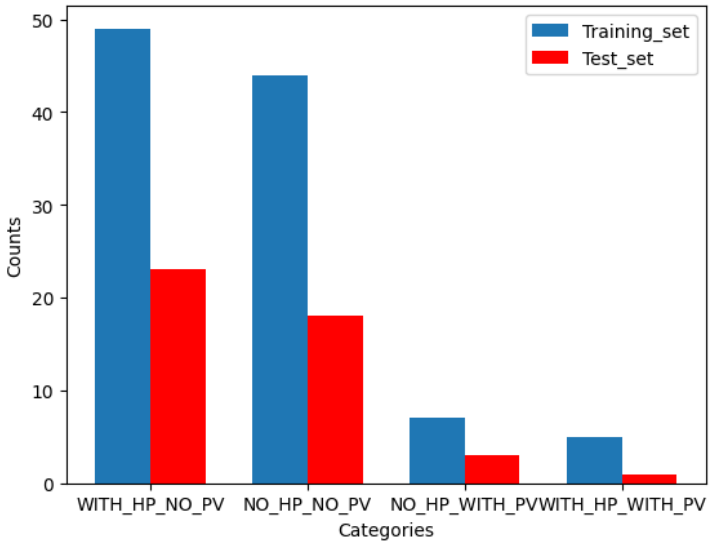


Figure 30 Distribution of the different labels in the training and tests sets.

scikit-learn library from python is used to train the models. Different algorithms are used as will be listed in x axis in Figure 31. Figure compares the accuracy of the different methods in prediction of labels. Moreover, to investigate the sensitivity of the accuracy to the features, the number of features were parties in two scenarios. First, all the features as explained previously were used. By that, even the features with a substantial missing datapoints were used in prediction. In a second scenario (Figure 31 right side), only the features that contained all the datapoints were used in training the model. To investigate the accuracy of the presented model, accuracy is abstractly defined as the number of correctly predicted labels divided by total number of labels to be predicted. If $\hat{y}_i$ is the predicted value of the i-th sample and y_i is the corresponding true value, then the fraction of correct predictions over n_{samples} is calculated with the equation below:

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$

Figure 31 shows that the modeling is an integrated task of the right choice of features and algorithms simultaneously. However, it can be concluded that with the right features, the algorithm will have minor impact on the outcomes. While, with inappropriate features, the entire method is prone to significant error. This proves the importance of the physics of the system in supervised learning when choosing the features. on the other hand, it shows limitation of this method to be easily scale-up because the model trained on a specific geographical, weather conditions, user behavior, and other physical attributes may not be easily used for other cases.

Funded by the European Union

Funded by the European Union. 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.

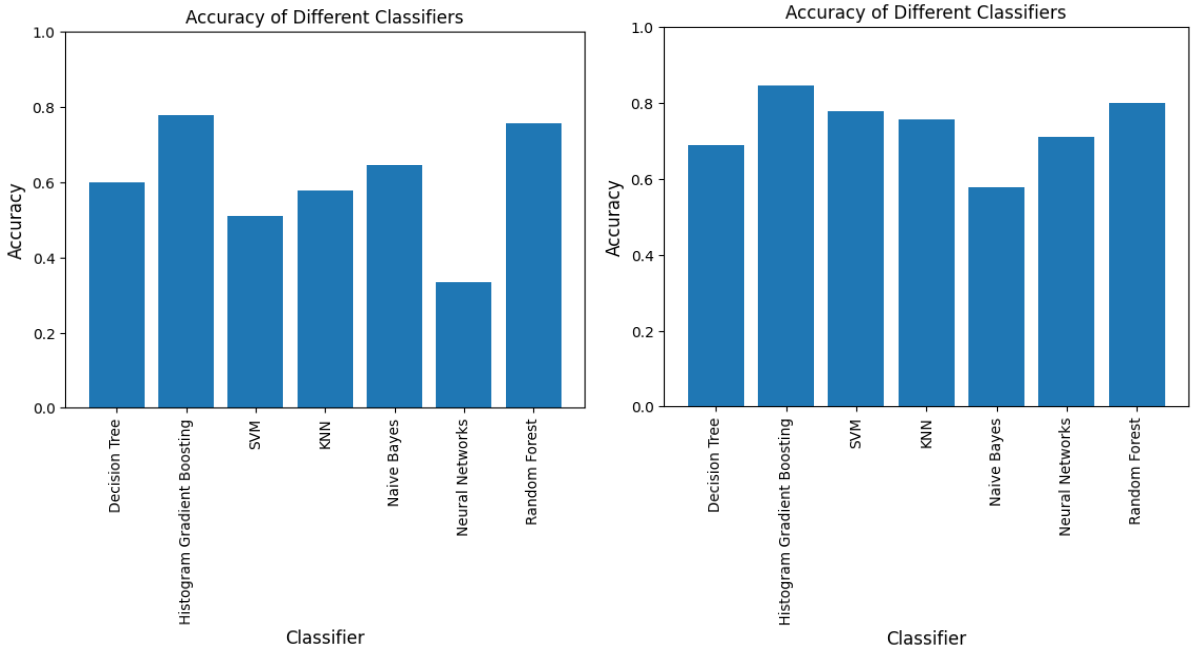


Figure 31 Sensitivity of the accuracy to the features and the ML technique. All the features from all months (with substantial missing points) were used in left. Only the second half of the year with complete datapoints in each month was used in training.

Examining the provided confusion matrix depicted in Figure 32 reveals a detailed account of the classification model's performance. The matrix shows that a significant error occurred when the algorithm wrongly predicted many labels as NO_HP_NO-PV. The diagonal elements correspond to accurate predictions, delineating instances where the model correctly identified labels (corresponding HP and PV installations). The model did not predict any label as PV without heat pump. This can be due to the lack insufficient training data for this label considering the distributions of the label (Figure 30). A meticulous examination of this confusion matrix provides a visual narrative of the classification performance, serving as a foundational guide for optimizing machine learning models. Confusion matrix shows how to approach for improving the model accuracy. For instance, 6 cases were predicted as with HP without PV while they had no HP installed. This error is not expected as the HP has very distinguishable impact on electricity time series specially when PV is not installed. These cases can be further studies to improve the model accuracy.

Funded by the European Union

Funded by the European Union. 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.

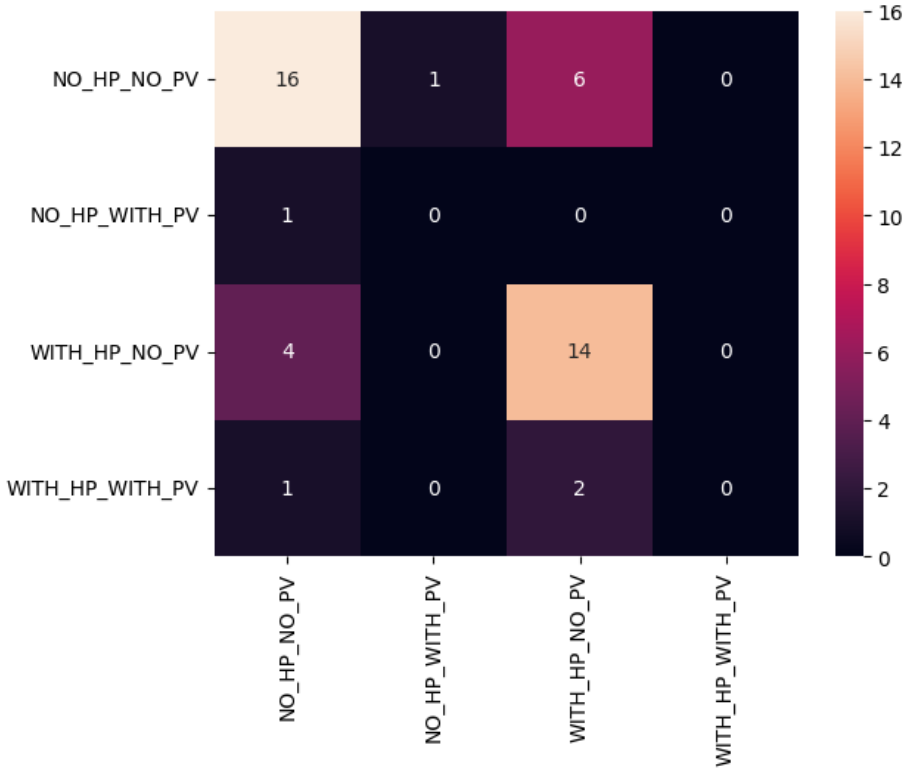


Figure 32 Confusion matrix for random forest method (y axis is the true label and x axis is the predicted label)

Shifting focus to the feature importance, Figure 33 unveils the varying degrees of influence exerted by individual features on the model's predictive prowess. Feature importance analysis involves examining the contribution of each input variable in shaping the model's output. Notably, the figure underscores the significance and impact of specific features, enabling the identification of pivotal contributors to the decision-making process. Features endowed with higher importance scores wield more substantial influence in molding the model's predictions, while those with lower scores possess comparatively diminished impact. This analysis proves instrumental in guiding feature selection, streamlining model complexity, enhancing interpretability, and potentially ameliorating overall model performance. By delving into the intricacies of feature importance, practitioners glean valuable insights into the internal mechanisms of the model, thereby facilitating judicious decision-making in the refinement and optimization of machine learning algorithms. Figure 33 shows that an interpretable order of importance in the selected features. Month 12 with possible highest heating loads stands out as the most important period to use for training a model. Yet, the other months can play a role in distinguishing the period during which PV is generating electricity and to predict buildigns with PV. Overall, this exercise again emphasized on the importance of considering the physics of the system when training the model with supervised ML techniques. Physics-based analysis can remarkably speed-up the procedure of choosing the right features for training the model.

Funded by the European Union

Funded by the European Union. 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.

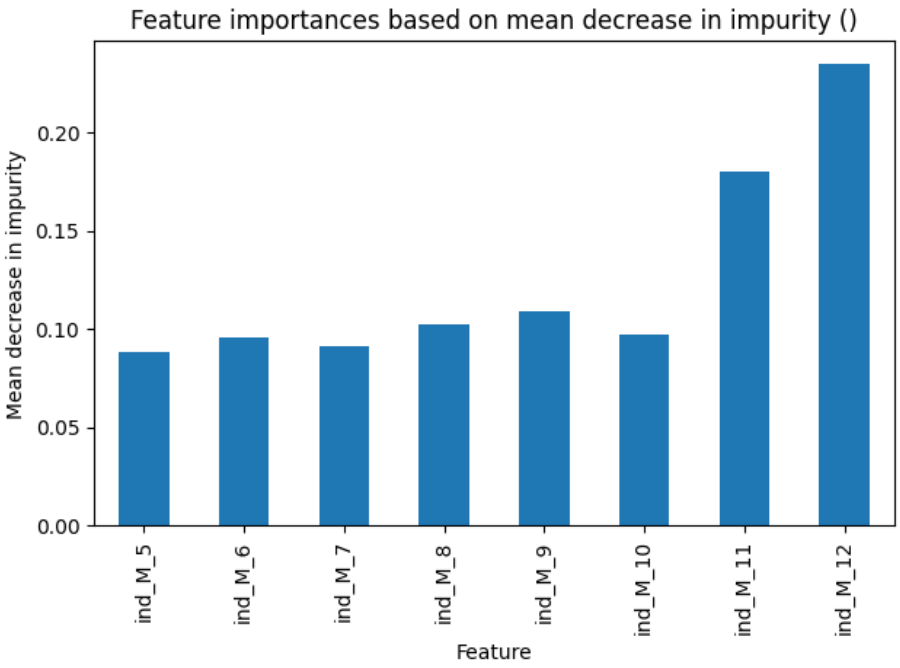


Figure 33 Feature importance for the developed model with random forest algorithm

Funded by the European Union

Funded by the European Union. 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.

4 Building Stock Synthetic Data Generation: A Belgian Use Case

To demonstrate a combined implementation of methodologies described in section *Load profiles on building level* and section *Synthetic data generation for tabular data* we use the standard building typologies as defined in the TABULA database for Belgium. We mainly focus on single family dwellings (with similar floor areas) and derive the building characteristics as defined in the database. These building characteristics include data on existing building insulation standards, building technical systems and current building consumption statistics.

4.1. Current Building Stock

The existing building stock characteristics are fed into an Input Data File (IDF), which is an ASCII file containing the data describing the building and HVAC system to be simulated. The IDF acts as the input file to run energy simulations in EnergyPlus. The synthetic dataset is created using parametric simulations by varying the following parameters.

1. External Wall U-values
2. Roof U-values
3. Window U-values
4. Lighting Flux Density
5. Electrical Appliance Ratings (Washing machine, dryer, dishwasher, refrigerator and television)
6. Gas boiler efficiency

These parameters are varied randomly between pre-defined minimum and maximum limits to generate 100 different building types. The parameters in the current stock do not adhere to the renovation standards and hence, represent the base case to compare the energy efficiency improvements. The simulations are run in a loop; each parameter is updated parallelly at the end of every loop. Through these simulations, we generate time-series electricity and gas consumption profiles for each building variant on a 15-minute basis. The consumption values are recorder in Joules and could be easily converted into kWh units (1 kWh = 3600000J). The yearly aggregated electricity and gas consumption values are validated against the national consumption statistics.

4.1.1. Time-series Electricity Consumption

The current building stock is designated as the base case. The time-electricity consumption patterns for the 100 building variants are depicted in Figure 34; the variations amongst the 100 profiles mainly occur due to different lighting efficiencies and appliance loads.

Funded by the European Union

Funded by the European Union. 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.

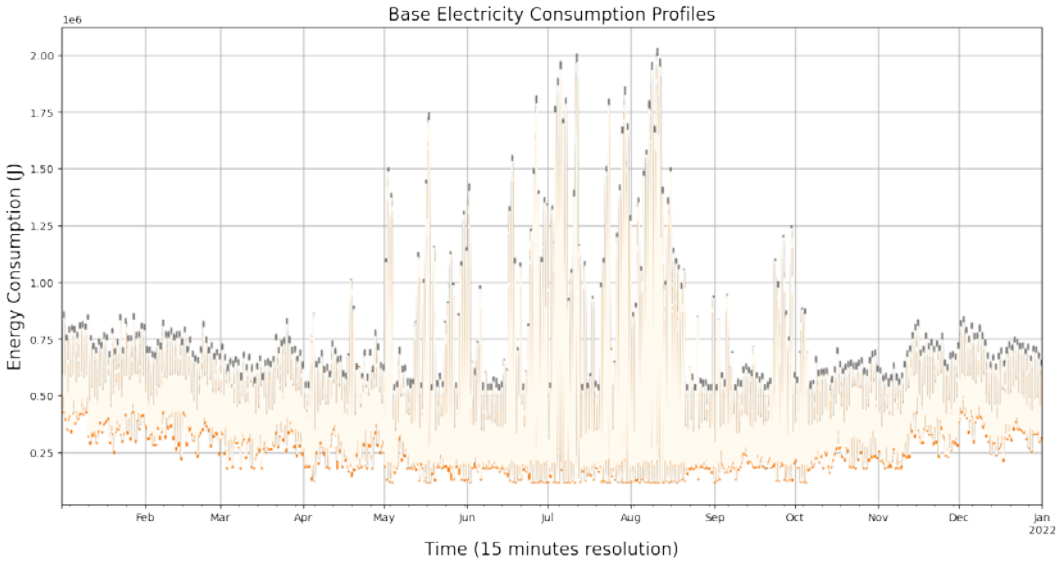


Figure 34 Time-series electricity consumption profiles for the formulated building stock comprising 100 different building types with no renovations.

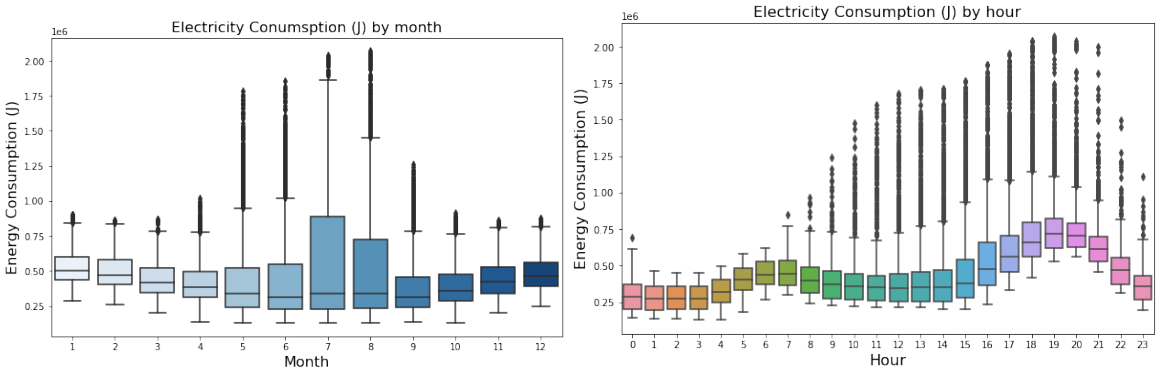


Figure 35 Electricity consumption classification of the building stock by month and by hour.

When comparing the electricity use patterns by month, there exist large number of outliers particularly during the summer months (between May and September) (Figure 35). This could be attributed to the fact that there occur random peak instances on hot summer days due to air conditioning requirements. When comparing the patterns by hours, the consumption stays within set limits from 1 AM up until 7 AM in the morning (Figure 35). More outliers appear in the later parts of the day due to increase appliance use with peaks occurring between 6 PM and 10 PM.

Each of these time instances act as classifiers, namely hour, dayofyear, month, quarter, dayofweek and year (Figure 36). When predicting electricity time-series, the hour classifier significantly influences the predictions as also seen in Figure 36.

Funded by the European Union

Funded by the European Union. 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.

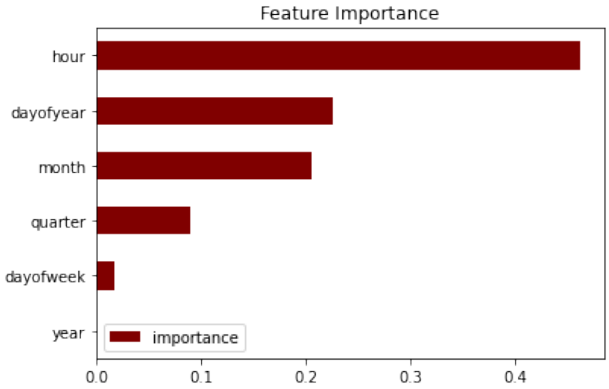


Figure 36 Feature importance of electricity time-series classifiers.

To demonstrate the use of synthetic time-series building data, we trained a machine learning time-series model using the XGBoost algorithm on an entire year of electricity consumption data (Figure 37). XGBoost (eXtreme Gradient Boosting) is an open-source algorithm that implements gradient-boosting trees with additional improvement for better performance and speed.

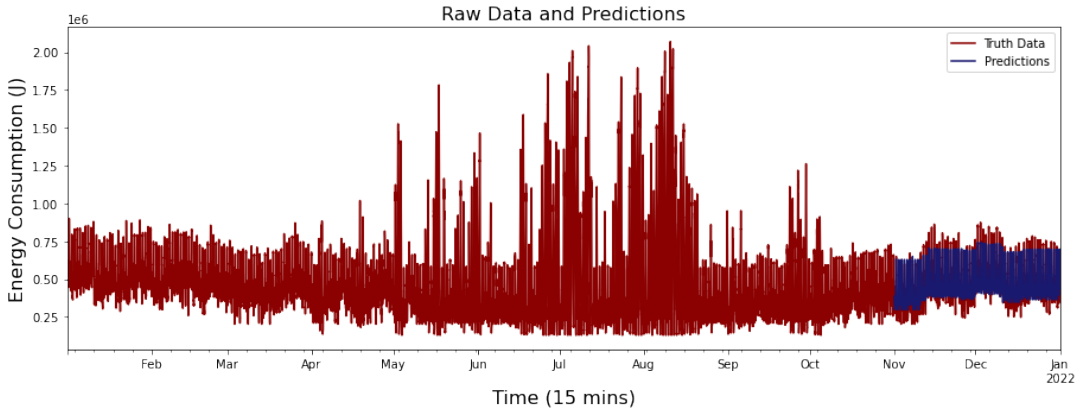


Figure 37 Time-series electricity prediction for the synthetic building stock using XGBoost algorithm.

We then tested the ML model to predict the electricity consumption from November 2021 to Dec 2022. The model can trace the time-of-use patterns with a root mean square error (RMSE) of 62875 J (or 0.175 kWh).

4.1.2. Gas Consumption Profiles

A similar process is used to obtain the gas consumption of the base building stock (Figure 38). The gas consumption profile comprises space heating, domestic hot water, and gas cooking range. The gas consumption patterns for the 100 building variants mainly vary in the peak occurrences due to the varying energy efficiency of gas boilers.

Funded by the European Union

Funded by the European Union. 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.

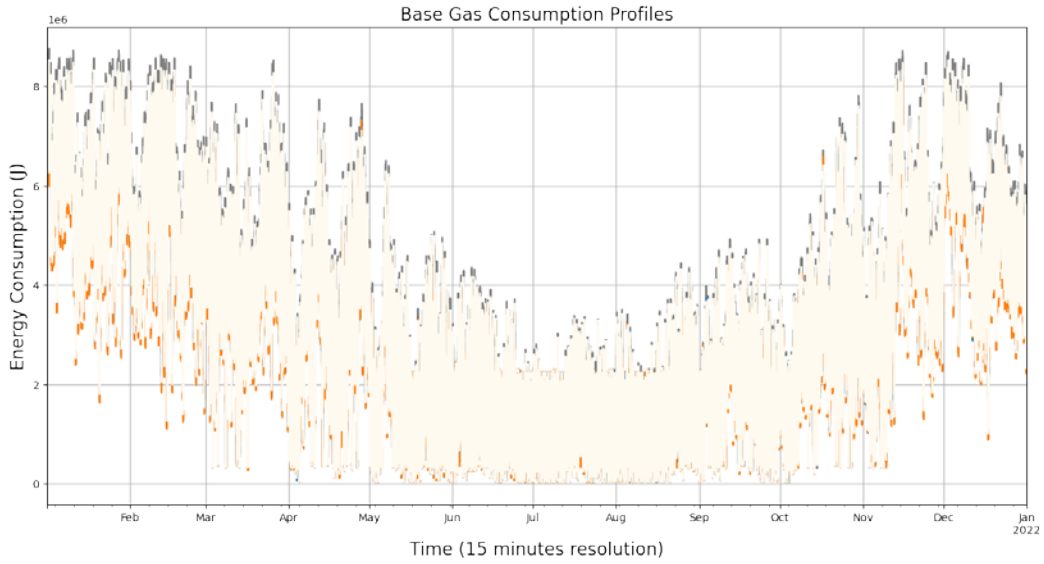


Figure 38 Time-series gas consumption profiles for the synthetic building stock comprising 100 different building types with no renovations.

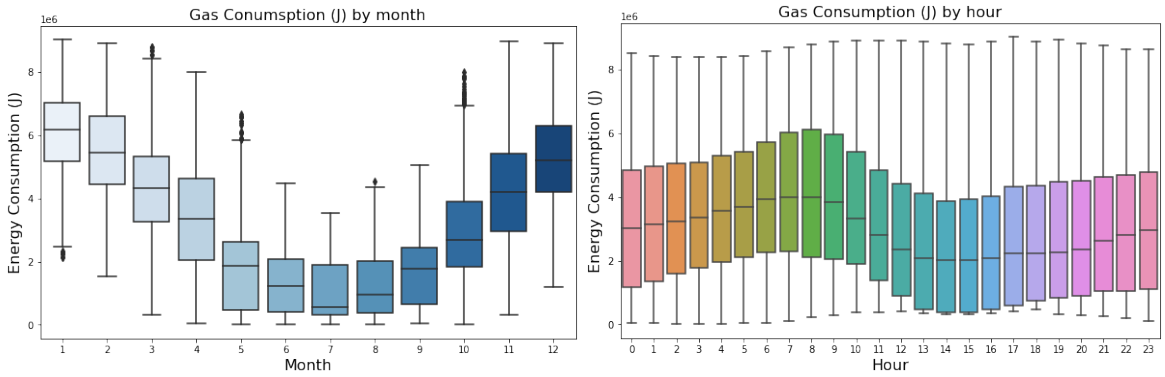


Figure 39 Gas consumption time-series classification of the building stock by month and by hour.

When analyzing the time-series classifiers for the gas consumption profiles by month, the outliers do not occur frequently mainly due to the delayed response of the heating system (Figure 39). When considering the hourly classifiers, there is always a minimum amount of gas use between 8 AM and 8 PM throughout the year (Figure 39).

Comparing the time features, month and dayofyear dominate the use of gas in single family dwellings as space heating mainly occurs during winter months (Figure 40). This is a significant finding and care should be taken when formulating time-series ML models for electricity and gas predictions.

Funded by the European Union

Funded by the European Union. 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.

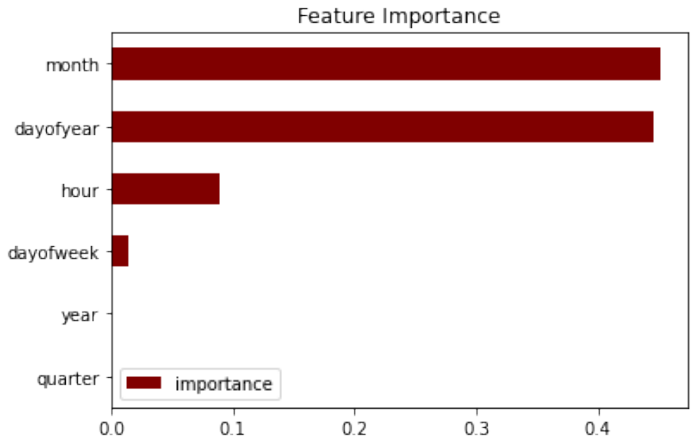


Figure 40 Feature importance of gas time-series classifiers.

We repeated the process of formulating the XGBoost ML model to predict the gas consumption from November 2021 to December 2023. The model is able to trace the variations in gas consumption with an RMSE of 1049543 J (or 0.291 KWh) (Figure 41).

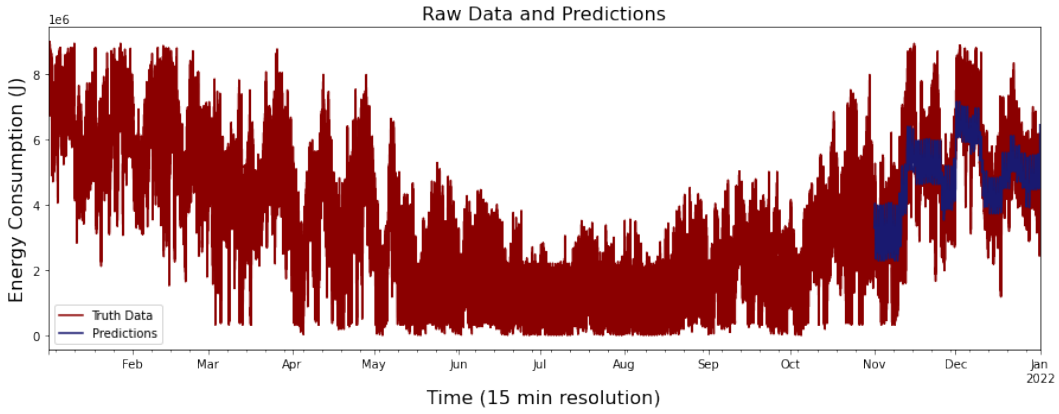


Figure 41 Time-series gas prediction for the synthetic building stock using XGBoost algorithm.

4.2. Renovated Building Stock

To create further building stock variants, the single-family dwelling is upgraded according to the latest renovation standards. Another loop is run to upgrade the aforementioned parameters and create a renovated building stock with 100 more variants.

When analyzing the electricity and gas consumption profiles, the range of variation in peaks decreases along with decrease in the aggregated energy consumption (Figure 42 and Figure 43). Feature importance rankings and the ML model display similar characteristics as those of the non-renovated building stock.

Funded by the European Union

Funded by the European Union. 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.

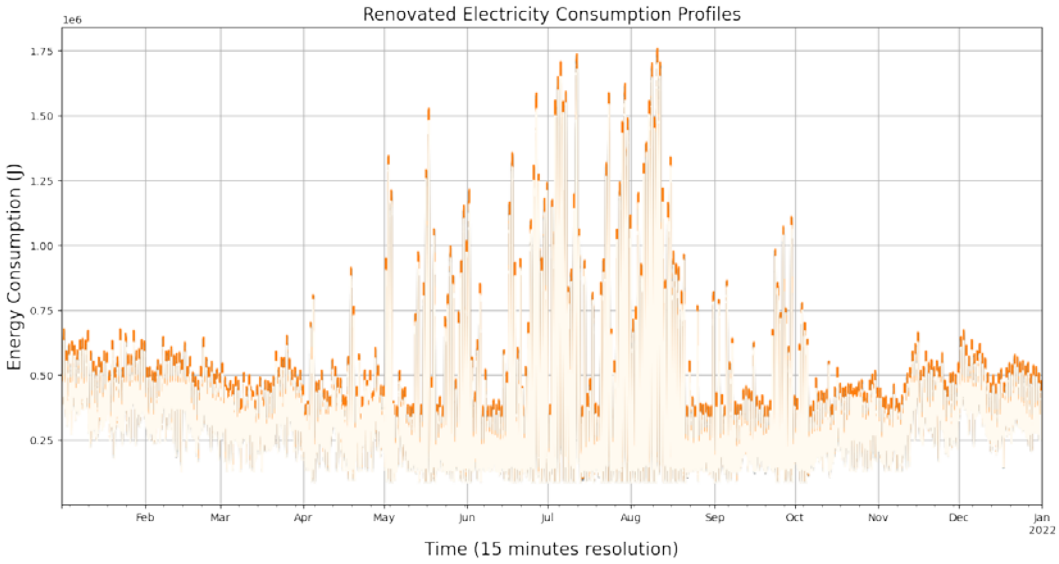


Figure 42 Time-series electricity consumption profiles for the synthetic building stock comprising 100 different building types with fabric renovations, boiler upgrades and energy-efficient electrical appliances.

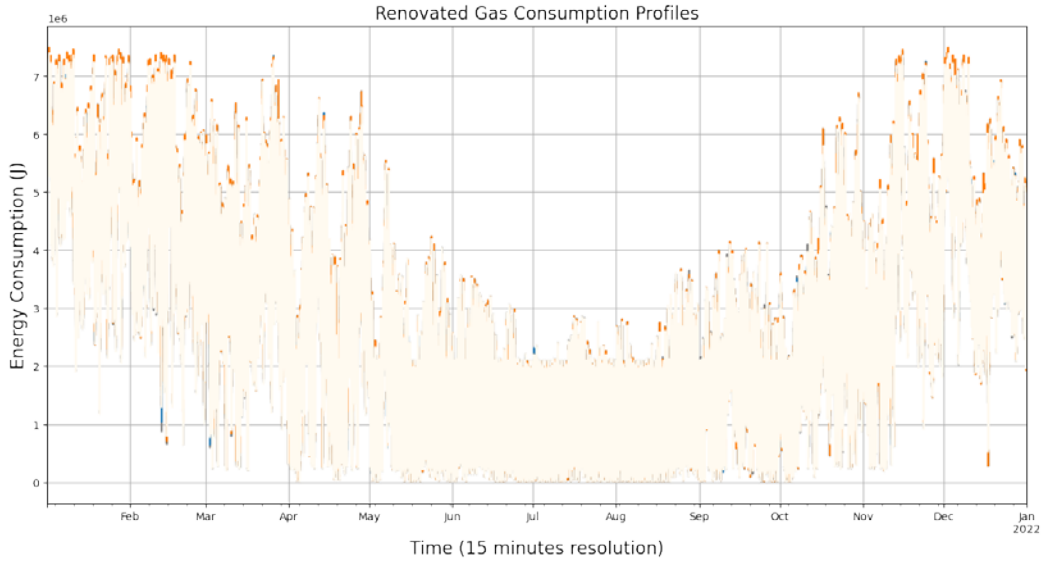


Figure 43 Time-series gas consumption profiles for the synthetic building stock comprising 100 different building types with fabric renovations, boiler upgrades and energy-efficient electrical appliances.

4.3. Renovated Buildings with Heat Pumps

To faciliate the decarbonization of the building stock, we created another renovated synthetic building stock (100 buildings) with the gas boilers being replaced by heat pumps (HPs). The HPs are air to water heat pumps with varying system COP.

Funded by the European Union

Funded by the European Union. 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.

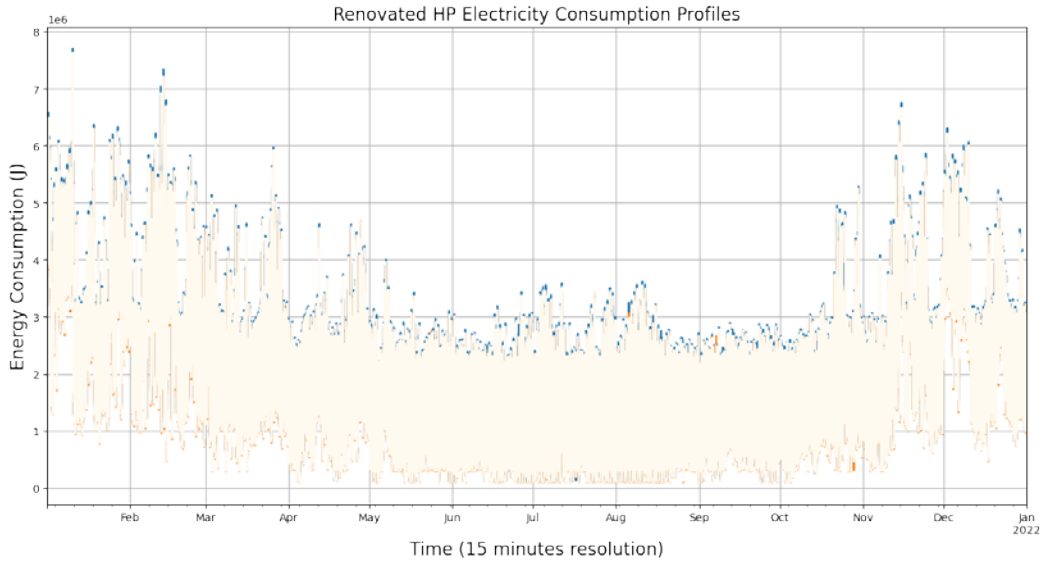


Figure 44 Time-series electricity consumption profiles for the synthetic building stock comprising 100 different building types with fabric renovations, heat pump installations and energy-efficient electrical appliances.

The time-series electricity profiles now resemble a combination of gas and electricity consumption profiles as of the previous renovated and non-renovated building stock with space heating, domestic hot water and cooking end uses being met by electricity instead of gas (Figure 44).

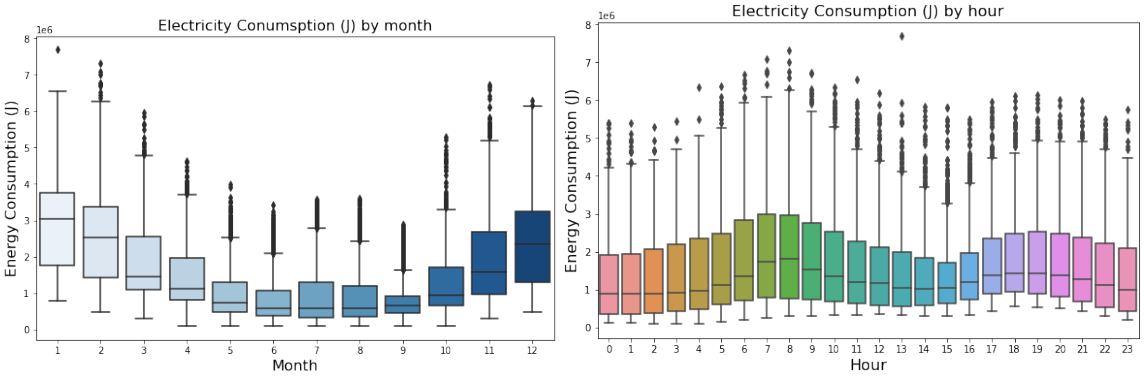


Figure 45 Electricity consumption time-series classification of the building stock by month and by hour.

A similar trend is observed when considering electricity consumption patterns by month and hour where significant number of peaks occurs in the time-series because of the transient response from the HP (Figure 45).

Month appears as the top influencing factor followed by dayofyear and hour indicating the time series patterns are dominated by space heating in the single-family dwelling (Figure 46).

Funded by the European Union

Funded by the European Union. 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.

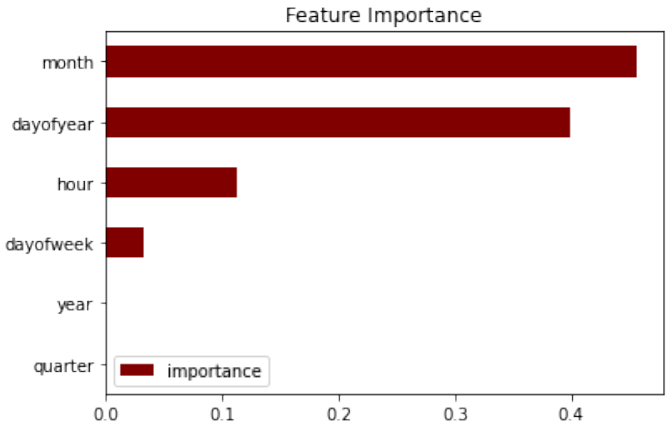


Figure 46 Feature importance of electricity time-series classifiers.

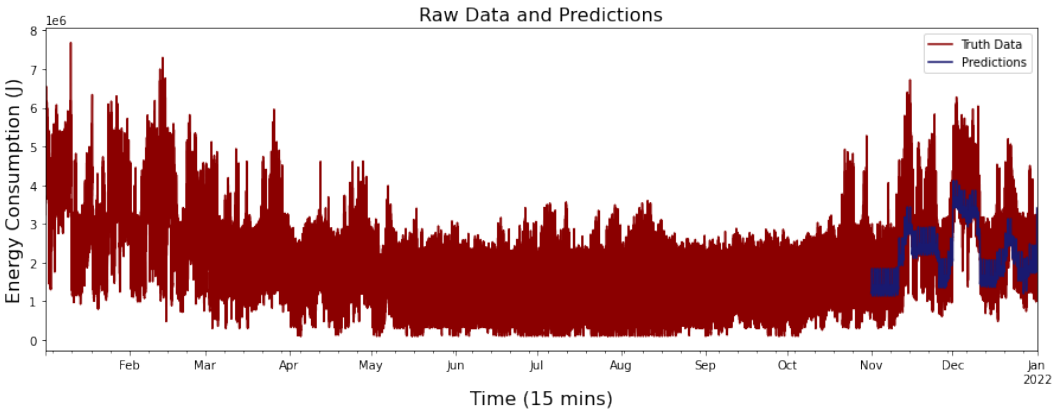


Figure 47 Time-series electricity prediction for the synthetic building stock using XGBoost algorithm.

The XGBoost ML model produces an RMSE of 954328 J (0.265 kWh) when predicting the electricity use from November 2021 to December 2021 (Figure 47). Such a model can effectively trace varying use patterns of HPs under different operating conditions.

The generated synthetic datasets will eventually be tested under varying weather conditions to test the climate resiliency of different renovation advisories. Furthermore, such datasets provide labels to classify any random time series patterns and hence, can be used to implement sophisticated ML classification techniques.

Outlook

This document summarizes the work done so far within the MODEATE project to achieve the goal of generating synthetic building data. In the further course of the project described methodologies will be tested on different data sources and compared to each other. If a chosen methodology turns out not to be efficient for a given dataset, the methodological approach might be changed in the future. At the end of the project the final methodologies will be presented together with a description of why these methods were ultimately chosen and the code will be made publicly available.

Funded by the European Union

Funded by the European Union. 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.

MODERATE

OUR TEAM

eurac research

Università Ca'Foscari Venezia

Politecnico di Torino

CTIC centro tecnológico

vito

ethink Energy Research

TU WIEN Technische Universität Wien Vienna | Austria

Köhler & Meinzer

FONDAZIONE links PASSION FOR INNOVATION

VEOLIA

enercoop GRUPO

ubik Geospatial Solutions

WÜRTH

IVE Instituto Valenciano de la Edificación

REHVA Federation of European Heating, Ventilation and Air Conditioning Associations

INSOMNIA

UCLouvain Louvain research institute for Landscape, Architecture, Built environment

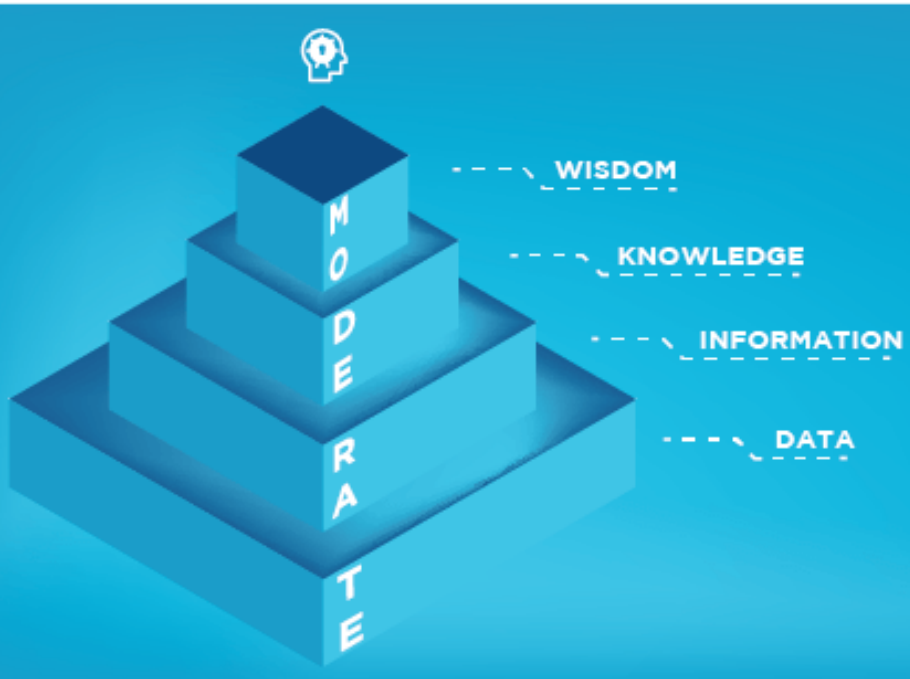
synavision Perfect Building Performance

See you online!

moderate-project.eu

@MODERATE_HE

MODERATE



MODERATE