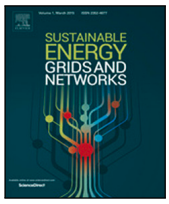
Sustainable Energy, Grids and Networks 41 (2025) 101622



Contents lists available at ScienceDirect

Sustainable Energy, Grids and Networks

journal homepage: www.elsevier.com/locate/segan





Explainable representative-days clustering on low-voltage grid meters and feeders, with noise-aware multi-objective Bayesian optimization, applied to grid-congestion events

Konstantinos Theodorakos [a,*], Oscar Mauricio Agudelo [a], Thijs Becker [b,c], Koen Vanthournout [b,c], Reinhilde D'hulst [b,c], Bart De Moor [a]

[a] STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Department of Electrical Engineering (ESAT), KU Leuven, Kasteelpark Arenberg 10, box 2446, 3001, Leuven, Belgium
[b] VITO, Boeretang 200, Mol, 3400, Belgium
[c] EnergyVille, Thor Park, 8310, Genk, 3600, Belgium

ARTICLE INFO

Keywords:
Low voltage grid
Representative days
Clustering
Time series
Optimization
Meteorological importance

ABSTRACT

Low-voltage grid (LVG) state estimations help in expansion planning and preventing congestion events. However, country-scale simulations pose high computational burdens. A solution that reduces calculation time is to cluster similar days, and only simulate the most representative day of each cluster. Using real-world, quarter-hour residential consumption time series from 925 meters, congestion event probabilities from 146 real feeders, 51 daily meteorological and 12 calendar features, we propose a novel end-to-end representative-days clustering framework. Along with k-medoids clustering, we use dimensionality reduction (kernel principal component analysis, factor analysis, ...) and pre/post-processing. To emphasize quarter-hour extremes, we apply dynamic data squeezing/expansion, based on the LVG consumption median. Our approach is scalable because dimensionality reduction compresses thousands of daily variables into up to 300 components, regardless of the amount of meters and exogenous features (weather, calendar, ...). Multi-objective Bayesian optimization for noisy functions, along with Sobol sampling, finds the hyperparameters that minimize the k representative-day reconstruction error compared to the full-year simulation. Explainability is achieved via tree ensemble classification on the decided clusters: ranking of the most important meteorological and calendar features, along with rule induction for future clustering decisions. We applied these techniques to successfully cluster all days of 2016 of the Flemish LVG (in Belgium). Our approach works well on both meter consumptions and feeder congestion event approximations. Photosynthetic radiation (visible spectrum and solar panel absorption wavelengths), water temperature, soil water volume/temperature (variables with a recent weather memory effect) and albedo were the most important meteorological factors, along with calendar features.

1. Introduction

The transition towards renewable energy makes it challenging to efficiently plan and operate the low-voltage grid (LVG). The fast and large-scale addition of electrical vehicles, heat pumps, and photovoltaic (PV) solar panels significantly increases peak demand. As a result, reinforcements need to be performed at a faster pace than is currently the case to avoid congestions [1–3]. If grid congestions become common place, it will hinder the energy transition, since further electrification (e.g. adding heat pumps) becomes impossible on the congested parts of the LVG, or load needs to be actively managed or curtailed [4,5]. Our objective is to cluster the days of a year and find the most representative day for each cluster, based on raw residential energy consumption time series, calendar cycles (season, weekend/weekdays, holidays, ...) and weather data (solar irradiation, precipitation, temperature, ...). The focus is finding similar LVG states by assessing how well we can predict quarter-hour congestion event occurrence from only the representative days (see Fig. 1 illustration).

Historically, the moment of peak demand was known. For Belgium this was a cold winter evening during a holiday. With the addition of extra loads, including ones that generate energy, congestion can shift to other moments. Some LV feeders become congested at a moment of

* Corresponding author.
E-mail address: konstantinos.theodorakos@esat.kuleuven.be (K. Theodorakos).

https://doi.org/10.1016/j.segan.2025.101622
Received 20 February 2024; Received in revised form 11 January 2025; Accepted 11 January 2025
Available online 18 January 2025
2352-4677/© 2025 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

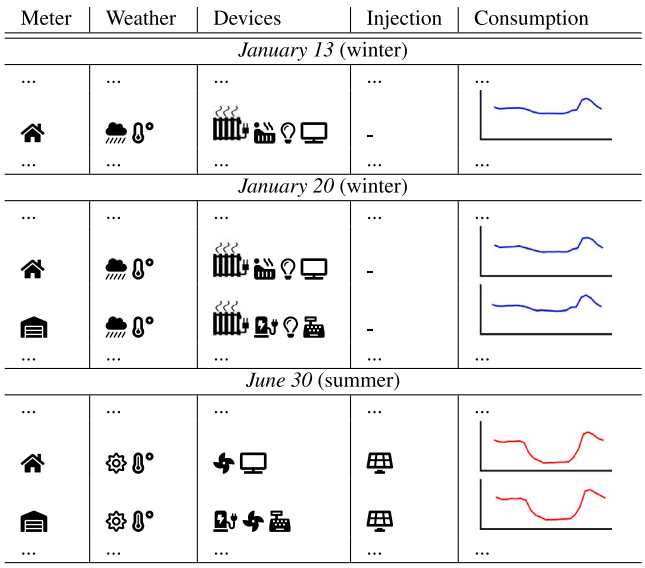


Fig. 1. Low-voltage grid (LVG) days-of-year clustering illustration: the goal is to cluster (group) the most similar days in regards to the LVG state. January 13 and January 20 should be clustered together, as they have similar: weather (cold and rainy), device consumption (space/water heating, lights), no energy injection and meter energy consumption in time (blue lines: household and warehouse). Summer days like June 30 exhibit different consumption patterns (warm weather with high irradiation, no heating devices, solar energy injection and the energy consumption has different peaks in time (red lines)). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

high PV injection, which happens in the middle of the day during spring or summer. Heat pump consumption depends strongly on temperature, which can shift the LV feeder peak to the coldest day, irrespective of whether this is a holiday. Electrical vehicle charging can also lead to peaks throughout the full year (irrespective of weather and seasons). We therefore need a fine-grained understanding of congestion issues on the LVG over a full year instead of a single moment. Since the LVG is sparsely metered, a good estimate of its state can only be obtained by simulations [6–8]. Simulating the state of the LVG at a country scale, over a full year and at high temporal resolution (hourly or quarter-hourly) is computationally intensive. For example, for Flanders it involves complex network structures with 3.5 million connection points of approximately 240,000 feeders, on which powerflows have to be run. To reduce the computational needs, we can cluster (group) the similar days of a year (see Fig. 1). Clustered days should have common characteristics (similar weather, electrical consumption, ...). By selecting appropriate clusters and only simulating the most representative day of each cluster, we can cover a full year with reduced computation times (e.g. a 183-day simulation is 50% faster than a 365-day simulation). Unfortunately, proper clustering and representative-day selection are difficult tasks that suffer from combinatorial explosion (see supplementary material for computational complexity).

1.1. Existing techniques and comparisons with previous work

The first representative time-period selection techniques were based on heuristics and optimization-based approximations [9]. Initial approaches did not attempt to select days but weeks [10]. With some recent exceptions [11], most previous works only focused on load profile segmentation and did not consider clustering in time (representative day/week selection) [9,10]. The focus was limited to generation and capacity expansion planning [12–15] but not on congestion events. Disruptive event detection is focused at the residential (appliance) level in [16–18], but not at higher levels of the LVG. Our focus is to cluster and select representative periods based on quarter-hour time steps, at two levels of the LVG: the meter and the feeder level. We combine residential consumption clustering at the LVG meter level, evaluated on probabilistic event detection clustering at the LVG feeder level, without any prior information or metadata about the feeders.

Outliers, missing and erroneous data are common occurrences in LVG datasets. It is usually caused by unforeseen failures and unplanned disruptions in data entry, collection, communication and gathering. Most publications do not mention their preprocessing approaches. Dimensionality and data reduction (via variable removal) along with k-means or k-medoids clustering were used in the past as a form of preprocessing, with an emphasis on costs [19,20]. We did not include costs, tariffs, or investment data as it was not a focus for our application. However, as with some recent comparative research [21], there was no explicit consideration about the effect of missing values, data errors or outliers. With our proposed framework, we address such issues both at the preprocessing and the clustering phases. In [18], the authors characterize their residential load event-detection approach as robust because of its parameter-adaptive characteristics and experimental performance. In an event-based appliance recognition paper [17], a Cepstrum filter that suppresses noise in time series is what implied robustness. For our case, we use extreme outlier removal and k-medoids which is robust to outliers. In addition to clustering robustness, we offer a comprehensive range of parameters and hyperparameters, along with dimensionality reduction options (kernel principal component analysis, factor analysis, ...) to the Bayesian hyperparameter optimization algorithm.

Dynamic Time Warping (DTW) was used to find representative days along with hierarchical clustering [12]. However, random variations between time series pairs in combination with time-warping may lead to temporal misalignments due to noise effects [22]. DTW may also violate the triangular inequality theorem as a distance measure [23]. Shape-based distance clustering approaches [21] show promising performance, however, they also apply time-shifting. In contrast, we do not disrupt time in our quarter-hour resolution data, as we are not only interested in identifying similar days but also in retaining the exact time information when congestion events happen. In our proposed framework, we include several fast-to-calculate and non-time-altering distance metrics as hyperparameters for the Bayesian optimization phase.

In a recent publication [13], the authors developed mixed-integer linear programming, input and cost-based approaches with k-means or k-medoids for representative-day selection. However, the experiments presented only up to 15 representative days. In our experiments, we examined from 1 up to 366 clusters/representative days, from highly granular (quarter-hour sampling time) real-world LVG meter and feeder time series. To examine associations between the clusters and the exogenous variables, we provide heatmap views of 51 meteorological variables, 12 calendar-based cycles/events, and finally, clustering explainability (with ranked feature importance and decision rule induction). To our knowledge, few publications account for weather or other exogenous variables along with energy demand and consumption data [11,24]. We apply multivariate time series data fusion and by extending the clustering into a classification setting, we can offer insights into the external variable influence.

1.2. Overview: Representative-days clustering framework

The top-level view of our proposed clustering framework is depicted in Fig. 2. In the preprocessing phase, we handle data issues with extreme-outlier removal and spline interpolation. Then we apply robust clustering with k-medoids. Bayesian optimization [25] selects: (1) the data dimensionality reduction method, (2) the distance metric that compares different days and (3) the k-medoids hyperparameters. Outputs are clustered days (groupings), along with representative days (cluster centers). We show that these representative days alone can

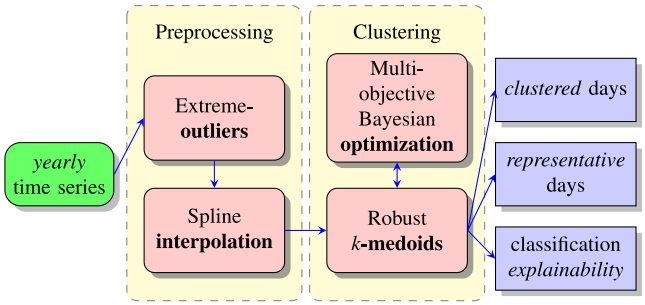


Fig. 2. Days-of-year clustering framework top-level view: yearly time series of low-voltage grid residential consumptions are preprocessed with extreme-outlier removal and spline interpolation on the missing values. The final goal is to cluster the days of a year with the multi-objective Bayesian-optimized and robust k-medoids and find the most representative day of each cluster. Applying classification techniques on the found clusters offers explainability.

approximate the full-year LVG state. Finally, we provide model explainability, by using the found clusters as classes and re-framing the problem into tree-ensemble classification.

For the clustering experiments, we use real-world data: 937 residential consumption profiles (quarter-hour sampling) from the region of Flanders in Belgium in the year 2016, 12 calendar features and 51 daily surface-level meteorological variables from the European Centre for Medium-Range Weather Forecasts (ECMWF) [26]. With our approach, it is also possible to assess the correlation between days, the weather and the LVG consumption patterns. **Contributions:**

- A novel end-to-end, scalable, and robust representative days clustering framework, resilient to noise and outliers.
- LVG meter and feeder clustering using noise-aware multi-objective Bayesian hyperparameter optimization [25]. The weighted geometric mean balances the multiplicative optimization emphasis among multiple timeseries error objectives.
- Clustered and representative days derived from a comprehensive set of variables: highly granular real-world LVG consumption time series, calendar-based events/cycles, and 51 daily meteorological features. Scalability is maintained by clustering on compressed daily data while retaining quarter-hour LVG level granularity within the objective functions.
- Clustering explainability and interpretability via ranked feature importance and decision rule induction.
- Introduction of several representative-day clustering benchmarks: “no clusters”, “calendar seasonal”, “repeat days” and “weekday-based”.
- We investigate how well our clustering approach can approximate the probability of congestion (under/over voltage and over-current) for a full year of 146 real low-voltage feeders. To our knowledge, no clustering publications exist that associate LVG data at the individual meter and feeder level concurrently.

This document is divided into 3 sections. Section 2 presents the framework’s inner workings in detail. Section 3 discusses the data and the experiments, LVG meter clustering, LVG feeder representative-day approximations and explainability. Finally, Section 4 provides concluding remarks and future research directions.

2. Methods

2.1. Overview

The goal is to perform *days-of-year clustering* and find the most *representative-day* of each cluster (see supplementary material for a background in clustering and distance measures). First, we use endogenous (i.e. quarter-hour meter consumptions) and exogenous variables (weather, calendar, ...), to cluster the 365 (or 366) days of a year. Afterwards, we define the representative day of each cluster, to reduce the amount of full LVG simulations needed.

Fig. 3 depicts the days-of-year clustering detailed view: using preprocessed residential electricity consumption and meteorological time series (temperature, rainfall, ...), we group days via robust clustering and find the most representative days of each group. The preprocessing phase deals with the outliers and missing values via rolling filtering and spline interpolation. After dividing the time series into days, Bayesian optimization finds the most performant dimensionality reduction method and k-medoids clustering hyperparameters. Outputs are the clustered days of the year (four days in the example for illustrative purposes), along with the representative days (cluster centers). Four representative days alone could approximate the yearly state of the LVG in this example. Furthermore, by reformulating the clustering problem as a classification problem, we can utilize the explainability of models like gradient-boosted tree classifiers. By applying tree-ensemble classification on the found clusters, model and clustering explainability is enhanced with: (1) cluster selection rule induction and (2) ranked input feature importance. The following subsections describe the blocks of Fig. 3 in detail.

2.2. Extreme-outlier removal

Dispersion is a statistical measure of the scatter or spread of a variable. A commonly used measure of dispersion is the standard deviation. *Robust* statistical methods provide statistics for data that may not be normally distributed and may be affected by outliers. We used the *InterQuartile Range (IQR)*, a statistically robust measure of dispersion for asymmetric distributions around their center [27]. Quartiles separate rank-ordered data into equal parts. The *median* (or quartile Q_2) is the value that separates ordered numbers into a higher and a lower half (50th percentile). The difference between the upper 75th (Q_3) and lower 25th (Q_1) quartiles is: $IQR = Q_3 - Q_1$. Tukey’s fence is an outlier detection method, that creates value boundaries between Q_3 and Q_1: [lower fence , upper fence] $= [Q_1 - f(Q_3 - Q_1), Q_3 + f(Q_3 - Q_1)]$, where f is a non-negative constant value. Observations outside these fences are considered outliers. Regular outliers are detected outside the $f = 1.5$ fence, whereas *extreme outliers* are outside the $f = 3$ fence.

The extreme-outlier removal step should be viewed as an optional component of the framework. In the dataset that we used, some of the examined time series contained large data gaps (of up to 24 h) due to data transmission delays. These data gaps were also followed by an (artificial) sudden spike in energy, which was the accumulation of all the missing values. To handle such data gap-spike issues, we removed the quarter-hour energy values that lie outside the extreme-outlier fence. However, if a dataset does not contain such erroneous data accumulation issues, the usage of the extreme-outlier fence should be carefully decided by the modeler to avoid removing inliers (data points that indeed belong to the time series and are not erroneously generated artifacts).

2.3. Rolling median filtering

To fill in missing values via interpolation, we used a robust trend estimate, the *rolling median*. The rolling (or moving) median is calculated using subsets of data and it is less susceptible to rapid time series anomalies than the commonly used moving average. Formally, given a time series $x = \{x_1, x_2, \dots, x_T\}$ of T data points, the rolling median $\tilde{x}_k$ at time-step k is: $\tilde{x}_k = median(x_{nk})$, with $k - 4s \geq 0$ and $k + 4s \leq T$. s is the stride of 7 days and $x_{nk} = \{x_{k-4s}, \dots, x_{k-s}, x_{k+s}, \dots, x_{k+4s}\}$ is the two-month rolling window of $n = 8$ samples (one sample per week). For the energy consumption use case, we used a data window of two months. From that window, we only use samples with a 7-day stride from the

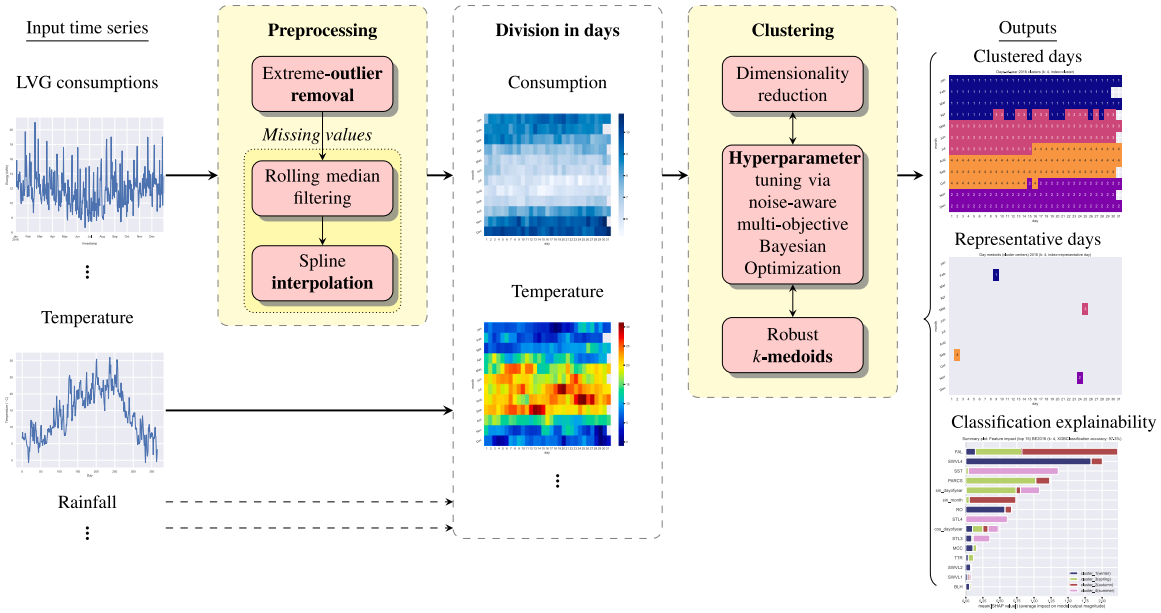


Fig. 3. Days-of-year clustering detailed view. Given yearly low-voltage grid (LVG) energy time series (quarter-hour residential energy consumptions from 937 m) and meteorological time series (51 variables like temperature, rainfall, ...), the goal is to group similar days via *robust clustering* and find the most *representative days* of the year. These representative days alone are enough to approximate the Flemish LVG state. *Preprocessing* phase deals with outliers and missing values via rolling median filtering and spline interpolation on the full yearly quarter-hour times series. After preprocessing the time series at a yearly quarter-hour level, we split them into days. Noise-aware multi-objective Bayesian optimization finds the most performant dimensionality reduction method and k-medoids clustering hyperparameters. Outputs: clustered days (four clusters in the figure), medoids/representative days of the year (cluster centers), and a ranking of the features that led to the clustering. Using the found clusters as classes, a tree-ensemble classifier provides ranked input feature importance for *explainability*. We use real-world data: 937 producer/consumer residential readings from Belgium in 2016 (region Flanders), 51 meteorological variables from the European Centre for Medium-Range Weather Forecasts (ECMWF) [26] and 12 calendar features (sinusoidal signals of the day of week/year, holidays, weekday or weekend, etc.).

k time-step that we want to fill in. In this way, we acquire electricity consumption values from exactly a few weeks before or ahead of time k. As an example, let us suppose that we have a missing value on Tuesday 22 November 2016 at quarter-hour 18:00–18:15. We use data from both the previous month and the following month (two month rolling window). We fill-in the missing quarter-hour with the median of 8 values: the values from 4 weeks before (18:00–18:15 from all the Tuesdays of the previous month: 25 October, 1 November, 8 November and 15 November 2016) and the values from the 4 weeks after Tuesday 22 November 2016 (18:00–18:15 from: 29 November, 6 December and 13 December 2016) of the exact same quarter-hour and day of the week (stride of 7 days).

2.4. Spline interpolation

The data (with a quarter-hour sampling frequency) had several consecutive missing values, spanning from 15 min up to 24 h, and the rolling median could not fill-in the full extent of the gaps. Since a typical linear interpolation was not sufficient in filling in the missing values properly, we used a 5th-order univariate spline interpolation. Smoothing splines are function estimates, a means of derivative-based smoothing and interpolation on noisy observations. The univariate spline interpolation [28], fits a smoothing spline $\mathbf{y} = spline(\mathbf{x})$ of degree $k = 5$, where $\mathbf{x}$ is a vector of N independent data points (noisy observations) and $\mathbf{y}$ is a vector of N dependent data points (smoothed observations). For the spline fitting, a positive smoothing factor $s_f \in [0, 1]$ chooses the number of knots (i.e. the number of predefined points that the spline passes through). With too large s_f the curve will be very smooth, but with too small s_f the curve will pick up a lot of noise. The number of knots starts with zero, and it is increased until the smoothing condition (weighted sum of squared residuals) is satisfied: $\sum_{i=1}^{N} (\mathbf{w}_i \times (\mathbf{y}_i - spline(\mathbf{x}_i)))^2 \leq s_f$, where $\mathbf{w}$ is a vector of N positive spline weights (all equal in our case).

2.5. Dimensionality reduction, exaggeration and scaling

Matrix decomposition is the mathematical operation of factorizing a matrix into a product of matrices. In our case, we use it for *dimensionality reduction* of the clustering dataset from $\mathbb{R}^{366 \text{ samples} \times 90\,000 \text{ features}}$ to $\mathbb{R}^{366 \text{ samples} \times 300 \text{ features}}$, where samples are days of the year (the year 2016 had 366 days) and features can be either electricity consumption values or weather/calendar data. With this data compression approach, the clustering remains scalable and consumes the same computational resources, regardless of the count of meters and the sampling frequency of the time series. The choice for 300 features was arbitrary, but we advise against decreasing it to less than 100 (since the given LVG dataset has 96 possible quarter-hour congestion moments per day). We applied *scaling* in the raw values to transform energy consumption and weather variables into similar value ranges while retaining their data distribution. Finally, we use *exaggerators* to shift focus and alter the raw data distribution. Meter consumption data belong in a different data distribution than the feeder event probabilities. The exaggerator function tries to bridge the gap between these two distributions, by squeezing and expanding values from a sinusoidal-like time series (of kWh consumptions) to a time series that contains values close to zero and some sudden (probability) spikes (see supplementary material for a visual comparison and a detailed discussion). By squeezing (reducing) low consumption values and stretching (increasing) high consumption values, the consumption timeseries data distribution becomes closer to the feeder time series that contain probabilities of congestion events (driven by consumption extremes). The exaggerator functions are dynamically centered around the daily LVG quarter-hour median (from all the meters). This way, no explicit expert-driven centering (based on specific kWh consumption ranges and time-of-day periods) is required. Tables 1 and 2 describe the relevant hyperparameters.

2.6. Noise-aware multi-objective Bayesian optimization

The main goal of the optimization is to minimize the yearly time-series reconstruction error of using only k clustered and representative

Table 1
Dimensionality reduction methods [29] used in the matrix decomposition hyperparameter of the Bayesian optimization (hyperparameter *decomposer* variable in Table 2).

Short name	Algorithm	Description
PCA	Principal Component Analysis	Linear dimensionality reduction using Singular Value Decomposition [30].
Kernel PCA	Kernel Principal Component Analysis	PCA extension where the original linear operations are performed in a Reproducing Kernel Hilbert Space (RKHS) [31].
Fast ICA	Independent Component Analysis	Decomposition of multivariate signals into independent non-Gaussian additive signals [32].
UMAP	Uniform Manifold Approximation and Projection for Dimension Reduction	Non-linear dimension reduction, assuming: (1) data are uniformly distributed on a Riemannian manifold, (2) Riemannian metric is locally constant and (3) the manifold is locally connected [33].
Truncated SVD	Truncated Singular Value Decomposition	Randomized truncated Singular Value Decomposition [30].
Mini-batch sparse PCA	Sparse PCA	Finds the set of sparse components that can optimally reconstruct the data [30].
FA	Factor analysis	Gaussian latent variables from a linear generative model [34].

days instead of the full 366 days of the year 2016. We used $n = 4$ objectives (two metrics × two datasets), on both the training and validation LVG meter subsets (to avoid overfitting on only a specific single set of meters) with the metrics: (1) Mean Absolute Error (MAE), and (2) Root Mean Square Error (RMSE). RMSE penalizes larger timeseries reconstruction errors strongly. MAE, in contrast, penalizes all errors with the same weight.

Due to the large scale of the annual LVG and the meteorological data, finding an optimal clustering model is difficult (see supplementary material for details on combinatorial explosion). Clustering models typically have an abundance of hyperparameters to tune: the number of clusters, clustering method, the distance metric, ...To find good clustering model hyperparameters, we followed a Bayesian Optimization (BO) [35] approach. There exist several families of optimizers [36] like random search, iterative mathematical solvers like Sequential Least-Squares Quadratic Programming, Evolutionary Algorithms (EA), ...We opted for BO because it usually performs better than other approaches on very few optimization iterations (we used 42 iterations, whereas e.g. EA works best in the order of hundreds up to thousands). Less optimization iterations require less computational time (clustering in our case) and fewer objective function evaluations. BO is a global optimizer, that defines prior and posterior distributions over objective functions (Gaussian process), using exploration and exploitation trade-offs on bounded parameter spaces. H is the tuple of hyperparameters to optimize (see Table 1 for a description of the matrix decomposition techniques used in dimensionality reduction and Table 2 for hyperparameter details):

$$H = \langle metric,\ method,\ init,\ decomposer,\ prescaler,\ postscaler,\ exaggerator,\ \#components,\ kernel \rangle \quad (1)$$

The number of clusters ($\#clusters$) can also be a hyperparameter, but we set it fixed $\in [1, 366]$ for experimental purposes. *Multi-objective Bayesian optimization* (MOBO) [37] finds hyperparameters for a black-box function with many objectives. Objective observations in a Pareto frontier are usually corrupted by noise. The *q-Noisy Expected Hypervolume Improvement* (qNHEVI) [25] extension is a Bayesian-optimal hypervolume maximization approach. MOBO qNEHVI utilizes parallelized gradient-based sample average approximations on both noisy and noiseless objectives. In our experiments, clustering with MOBO qNHEVI [25] generalized better versus the non-noisy MOBO variant or the single-objective BO [35]. Before the 20 iteration MOBO qNEHVI, we initialize the search with 22 random *Sobol samples* [38]. Sobol sequence is a quasi-random, low-discrepancy series of numbers that proportionally partitions our multidimensional hyperparameter space of 10 dimensions. This approach samples the search space in a more equidistributed manner than uniform random sampling, with very few samples: (10 hyperparameter dimensions +1)×2 samples per dimension = 22 samples. Formally, given an S-dimensional unit hypercube $I^S = [0, 1]^S$ and a real integrable function f over I^S, Sobol sampling tries to construct a sequence x_n of n samples, such that: $\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} f(x_i) = \int_{I^s} f$. In our experiments we also tried uniform and random normal sampling but they were not performing as well as Sobol sampling on 22 samples. Finally, to choose the single best solution over all four minimization objectives, we used the *weighted geometric mean* [40] (with equal weights). The geometric mean has an implicit proportional and percentage-change nature compared to the arithmetic mean. Given a sample of n objectives o_n and weights w_n, the weighted geometric mean is: $\bar{o} = \left(\prod_{i=1}^{n} o_i^{w_i}\right)^{1/\sum_{i=1}^{n} w_i}$. Unlike the arithmetic mean, the geometric mean's multiplicative nature puts higher overall penalties on under-performing objectives.

2.7. Representative-days objectives optimization

The main goal of clustering is to find which days of a year can be repeated, such that we have minimal energy consumption variance in time, at the LVG **meter level** (Algorithm 1). In other words, given hyperparameters H and LVG meters M, we want to find a good enough K-days combination of: (1) groups (K clusters) of similar days D_K of a year D (Line 2) and (2) the representative day r_k of each cluster k that could replace all the other days d_k of the group D_k (Lines 5–6) with minimal consumption differences versus the full year (366-days) (Line 8). The noise-aware Bayesian optimization tries to find the best possible hyperparameters H, that minimize the weighted (Line 7) geometric mean $\bar{o}_K$ of the objective functions o_n (Line 9). The objectives utilize the full LVG dataset granularity since the clustering evaluations are done at the quarter-hour resolution level (Line 8) on the training M_{train} and validation M_{val} meters. The impact on execution times is small because the objective function is calculated only a few times, as many as the number of Bayesian optimization iterations (for our case: 22 pseudo-random Sobol iterations for sampling of the hyperparameter search space + 20 MOBO qNEHVI iterations). k-medoids execution speed is not affected, since the clustering algorithm works on a smaller dataset after applying dimensionality reduction ($\in \mathbb{R}^{366\ days \times 300\ components} \approx \mathbb{R}^{110{,}000}$), a great reduction versus the full LVG dataset ($\in \mathbb{R}^{937\ m \times 366\ days \times 96\ quarter-hours} \approx \mathbb{R}^{33{,}000{,}000}$). This way, the clustering step (which is the most computationally intensive) remains scalable, regardless of the size of the LVG dataset.

Table 2
Bayesian optimization hyperparameters [29,39] and bounds.

Hyperparameter	Values	Description
#clusters	integer $\in [1, 366]$	Number of clusters.
Metric	{cosine, Euclidean, Manhattan, BrayCurtis, Canberra, Chebyshev, Mahalanobis, Minkowski, SQEuclidean, SEEuclidean}	Sample-to-sample dissimilarity/distance metric.
Method	{alternate, PAM}	Clustering method.
Init	{random, heuristic, k-medoids++, build, benchmark}	Clustering initializer: 'heuristic': medoids are points with smallest distance sum, 'k-medoids++': based on 'k-means++', 'build': greedy (non-robust), 'benchmark': repeat consecutive days baseline.
Decomposer	{PCA, Kernel PCA, Fast ICA, UMAP, Truncated SVD, Mini batch sparse PCA, Factor analysis}	Matrix decomposition technique for dimensionality reduction.
Pre-scaler, post-scaler	{None, MinMaxScaler, MaxAbsScaler, RobustScaler, StandardScaler, QuantileNormal, QuantileUniform, L1 normalizer, L2 normalizer, MaxNormalizer}	Raw kWh timeseries value scalers.
Exaggerators	{None, fixed, median, expit, exp, log_sigmoid, softplus}	Functions that squeeze and/or expand raw kWh timeseries values.
#components	integer $\in [10, 300]$	Number of decomposition components.
PCA kernel	{linear, polynomial, Radial Basis Function (RBF), sigmoid, cosine}	Kernel function (only with Kernel PCA).

Algorithm 1 Multi-Objective: Representative-days fitness

function K_RepresentativesFitness(hyperparameters H)
  $D_K, R_K \leftarrow k$-medoids(H, K, D, M_{train}) ▷ Cluster year
  for cluster $k \in K$ **do**
    for clustered days $D_k \in D_K$ **do** ▷ Days-group D_k
      for day $d_k \in D_k$ **do** ▷ Replace days
        $d_k \leftarrow r_k$ ▷ Representative-day r_k (kWh)
  $w_n = \{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\}$ ▷ Weights for n objectives
  $o_n \leftarrow error(K\text{-days, 366-days})$ ▷ MAE & RMSE $\in M_{train,val}$ (kWh)
  $\bar{o}_K \leftarrow \left(\prod_{i=1}^{n} o_i^{w_i}\right)^{1/\sum_{i=1}^{n} w_i}$ ▷ Weighted geometric mean
  return $\bar{o}_K$ ▷ Yearly quarter-hour kWh fit

2.8. Probability divergence of LVG feeder congestion events

The second goal of this research work is to find which days of the year can be repeated for the approximation of congestion LVG events, at the **feeder level**. Event occurrence can be described by probability time series of four event types: over-voltage (OV), over-current (OC), under-voltage (UV) and "any" (all the event types). For each quarter-hour of the year, we have simulated the probability of congestion using a Monte Carlo-based, statistical three-phase backward–forward sweep power flow calculation. An example of the output we are trying to reproduce is shown in the Results Section 3.3, where we show the typical result of an LVG feeder that has the highest probability of congestion on winter evenings. In our dataset of real feeders, we have a variety of congestion scenarios, such as over-voltage because of PV injection, etc. Congestion probabilities are calculated as follows. In the first step, we draw samples from the probability distribution of the consumption profile, for each house. This is done by sampling real electricity consumption time series from similar connections for which smart meter data is available. How this can be done is explained in [6]. We draw 200 samples for each connection on the LV feeder. Each set of samples (one per connection) is used in a powerflow simulation, to calculate the currents and voltages. If there is congestion at any point in the feeder, we take that feeder to be congested for that quarter-hour. After doing this 200 times, we estimate the probability of congestion as the percentage of times congestion occurred.

Day-groups found from meter time series clustering (containing residential energy consumptions) can be applied to feeder time series (containing LVG event probabilities), as the state of the LVG is mostly determined by local consumption patterns. k-medoids not only finds a good set of day groupings D_K for a year but also a representative-day r_k for each cluster grouping D_k. To approximate the 365 days of a year using only K days, we replace the days of a cluster with its most representative day. For example, to approximate a full year with the calendar seasonal clustering benchmark we have $K = 4$ clusters: (1) the days of a year are grouped into four clusters based on the season (winter, spring, summer, autumn) and (2) each day in a cluster is replaced with the most representative day of the season (i.e. the calendar solstice or equinox of that season). In this way, one only needs four days of data/simulations to approximate a full year. To approximate the quarter-hour probability of the occurrence of an LVG event at the feeder level, we applied this yearly approximation approach using clusters and representative days found from k-medoids clustering at the meter level. In other words, to reduce the number of powerflow Monte Carlo simulated days needed to approximate a full year, we can readily apply the clusters D_K and the representative days R_K found from the meter consumption time series directly to feeder event probability time series. The benefit of this approach is that we can group feeder time series by day, without any data or knowledge about the LVG feeders. In other words, we did not cluster the feeder probability time series directly or use any feeder metadata. We used the same clusters/representative days found from the meter consumption time series (Fig. 4 presents the flow of the full framework).

To assess the event-probability dissimilarity between the approximated days and the full year, we used probability divergence metrics. A commonly used metric that compares two probability vectors is the Kullback–Leibler (KL) divergence [41,42]. KL divergence is a statistical distance measure that assesses how much the probability distribution q differs from the reference distribution p. In our case, the reference distribution p_{366} for a feeder is the quarter-hour probability of an event for all the days of 2016 and q_K is the approximated year using K clustered and representative days of the same feeder (created by repeating the same representative-day on each cluster). For discrete probability distributions p_{366} and q_K defined in the same sample space $\mathcal{X}$, the KL divergence D_{KL} is:

$$\mathrm{D}_{KL}(p_{366} \parallel q_K) = \sum_{x \in \mathcal{X}} p_{366}(x) log\left(\frac{p_{366}(x)}{q_K(x)}\right) \tag{2}$$

where $\mathrm{D}_{KL} \geq 0$ is the relative entropy from q_K to p_{366}. An extension of KL divergence is the Jensen–Shannon divergence (JSD) [43]. JSD measures the probability divergence between the grid event probability vector p_{366} (of all the 366 simulated days $\in \mathbb{R}^{146\ feeders \times 366\ days \times 96\ quarter-hours}$) against the approximated probability vector q_K (generated using only K simulated days via k-medoids representative-day clustering):

$$JSD(p_{366} \parallel q_K) = \sqrt{\frac{\mathrm{D}_{KL}(p_{366} \parallel m) + \mathrm{D}_{KL}(q_K \parallel m)}{2}} \tag{3}$$

where $JSD(p_{366} \parallel q_K) \in [0, 1]$ is the information radius (IRad) or the total divergence to the average, m is the point-wise mean of p_{366} and q_K, and D_{KL} is the Kullback–Leibler (KL) divergence. The advantages of JSD are: (1) it is a smoothed version of D_{KL}, (2) always has a *finite value* $\in [0, 1]$ and (3) it is symmetric in p_{366} and q_K.

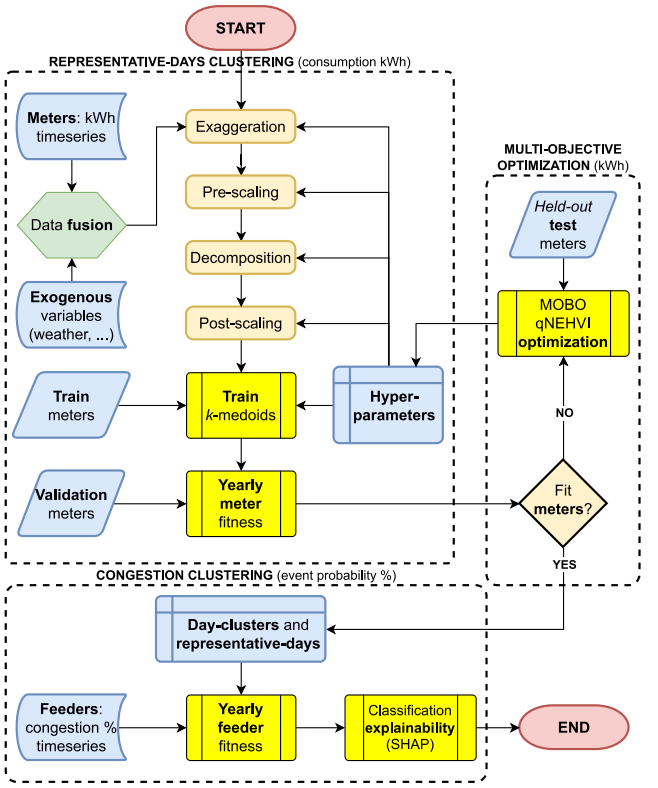


Fig. 4. Flow chart of the representative-days clustering framework. Three main phases: (1) representative-days clustering, (2) multi-objective optimization and (3) congestion clustering. In *representative-days clustering*: raw meter timeseries are fused with exogenous variables, then modified (with exaggeration, decomposition and scaling) and finally the k-medoids representative days are assessed against the full-year simulation. In *multi-objective optimization* the noise-aware Bayesian multi-objective optimization minimizes the yearly reconstruction error of the held-out test meters. Finally, in *congestion clustering* the optimized day-clusters/representative-days are assessed against the yearly (full 366-day simulation) feeder congestion event probabilities and Shapley additive values (SHAP) provide clustering explainability.

2.9. Clustering benchmarks

To assess and validate the k-medoids performance, we introduce several clustering benchmarks to compare against:

- "no clusters": all 366 days are put into one group ($k = 1$), with day 183 as the representative (mid-year, 1 July 2016).
- "calendar seasonal clustering": the days of the year are split into four groups, one per season ($k = 4$, with the seasonal solstice or equinox as representative day).
- "random clustering": we split the days of a year into random clusters, with a random day as a representative day within a cluster (varying k).
- "repeat days" clustering: we repeat the same day for 2, 3, ..., up to 25 consecutive times in sequence. The representative day is the first day of each repeating cluster. For our experiments with $k \in \{15, 30, 60, 90, 180, 210\}$: we repeat the same day from two consecutive times ($k = 210$ clusters), up to 25 ($k = 15$). Since it was the best-performing dynamic baseline, it was added as the "benchmark" k-medoids initialization hyperparameter.
- Weekday-based clustering:
  - "one-a-week"($k = 53$, the number of weeks in 2016): every week is a different cluster, with Wednesday set as the representative day of each cluster.
  - "two-a-week" ($k = 106$): two clusters per week (Monday to Friday and Saturday to Sunday), with Monday and Saturday as representative days.
  - "three-a-week" ($k = 158$): three clusters per week, with Monday, Thursday and Saturday as representative days.
  - "four-a-week" ($k = 210$): four clusters per week, with Monday, Thursday, Saturday and Sunday as representative days.

2.10. Explainability: Decision and gradient-boosted trees

A Decision Tree (DT) is a supervised Machine Learning method used for classification and regression. DT are tree-like models, that contain conditional statements over features (e.g. if the temperature is high and the electrical consumption is high, classify day as summer-day, else as winter-day). DT are white box models that require little data preparation and are easy to visualize and interpret. We use classification DT to provide Boolean logic-based explanations (rule induction) about the days-of-year clustering model decisions, against input meteorological variable ranges. In addition to DT, we use Gradient Boosted Tree (GBT) classification. Gradient boosting is a machine learning technique that combines multiple weak prediction models like DT in an ensemble. Tree-based models such as decision trees and gradient-boosted trees offer explainability by ranking the features that provide the largest information gain. We use GBT to provide ranked input feature importance about the clustering labels with Tree SHAPley values. Tree SHAPley values [44] is a feature importance extension for tree ensembles, that combines local statistical explanations with game theory. Combinations of local feature interaction effects provide global model interpretability. There exist machine learning frameworks, like scikit-learn [29], that offer a plethora of classification techniques. The reason that we opted for GBT instead of a different classification approach (like a Deep Neural Network or Random Forests) is that the SHAP explainability ranked feature importance [44] was originally designed with native support for tree ensembles. The authors in [44] also mention that GBT along with SHAP can offer more comprehensive explainability compared to linear models and showcase better explainability compared to Random Forests.

3. Results and discussion

3.1. Description of the data and the experiments

We focused on Belgian Flemish consumption profiles from 2016 to concurrently examine two aspects of the LVG: (1) the meter level with residential time series readings and (2) the feeder level with congestion event probabilities (quarter-hour sampling time, probability of over/under-voltage and over-current events).

For the residential consumption data (locations unknown), we applied feature engineering on the injection and offtake values. Specifically, to account for the energy that PV systems generate, we used the endogenous variable: LVG meter energy consumption = offtake (*from* the grid) - injection (*to* the grid). We augmented our datasets with 51 meteorological daily exogenous variables, which we acquired from the European Centre for Medium-Range Weather Forecasts (ECWMF) [26] (see supplementary material for all the meteorological variables: temperature, precipitation, pressure, ozone, solar radiation, ...). The weather was sampled from a central Belgian location (city Andenne, rural region). We did not apply any preprocessing on the weather variables, other than the 4D-variational assimilation from ECWMF. Finally, we added some calendar-based features (see supplementary material): weekend/weekday, holidays, day/week/month of the year, etc.

Regarding the yearly feeder congestion probabilities (quarter-hour time series $\in \mathbb{R}^{146\ feeders \times 366\ days \times 24h \times 4\ samples/hour}$), we only had simulation results for 365 days. Since 2016 is a leap year and has

366 days, for our experiments we repeated the 365th day (as the last day). Each of the 146 feeders had probability time series for four feeder event types: over-voltage (OV), over-current (OC), under-voltage (UV) and "any" (all event types). The feeders were manually selected throughout Flanders (locations unknown). Even though these feeders were intentionally selected to have a higher probability of congestion events than a typical one, the probabilities contained mostly zeros (or values very close to 0%). The main reason for this sparsity is that congestion events happen very rarely. The vast majority of feeders are today sufficiently strong to support Flanders' LVG connections without risk of congestions.

As an ablation study, we compared: no clustering, clustering benchmarks, random clustering, k-medoids clustering with or without exogenous variables and optimized k-medoids clustering with 20 multi-objective Bayesian optimization iterations. We ran experiments with one up to 366 clusters and representative days. Compared to a full year of simulations (366 days), 180 representative days require ≈50% calculation time, and 30 representative days only ≈12%. We did not run any experiments with other clustering algorithms, given that k-medoids fulfills quite well all our requirements for representative-days-of-year clustering [45]: (1) the desired number of clusters should be an input parameter, (2) the clusterer should be able to find cluster centers, (3) cluster centers (representative days) should be actual data points (specific days) and not virtual locations in a Euclidean space and (4) ability to choose from multiple distance metrics. Other than the typical Manhattan and Euclidean distance, k-medoids can utilize pair-wise distance metrics, which provide a very large variety of options for the hyperparameter search. Even though we did not compare against other clustering algorithms, we define benchmarks for the final clustering assessment and validation.

3.2. Data issue handling

Some of the meters that we had at our disposal, had a very high amount of missing values. From the datasets of Flanders with 937 m, several meters had ≈20% to 50% missingness (zeros). We removed the meters that had more than 75% missingness, so we ended up with 925 m available for the clustering experiments. We divided those meters into three datasets: training (50%), validation (20%), test (30%) subsets. There was an imbalance in consumption through time: most meters exhibited similar consumption patterns, but some had very high variance. To split up the meters evenly in terms of variance between the different subsets, we applied stratified sampling: we proportionally sampled the meters from three strata (buckets) based on meter consumption variance through time $log(STDev(\{x_{t_{start}}, \dots, x_{t_{end}}\}))$, where x_t is the energy consumption (kWh) of meter x at timestep t, t_{start} is the first quarter-hour of 1 January 2016, t_{end} is the final quarter-hour of 31 December 2016 and $STDev$ is the standard deviation. This procedure reduces the meter-type imbalance between the train, validation and test subsets, and the log transformation improves the division evenness.

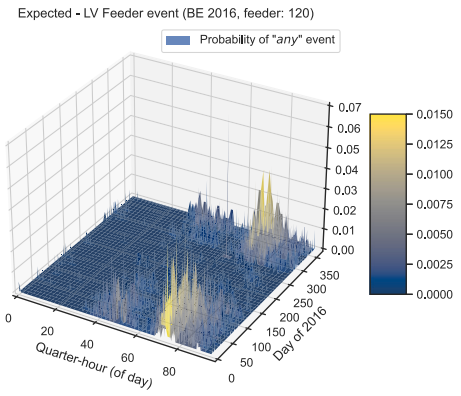
3.3. LVG representative days

Fig. 5 shows an example comparison of the full 366 days feeder event simulation data against different representative-day selection techniques: (1) the seasonal benchmark, (2) random clustering, and (3) optimized k-medoids clustering (60-day approximation). The seasonal approximation (Fig. 5(b)) is generally close to the full-year (366-day simulation) feeder congestion event probabilities (Fig. 5(a)) but it is very coarse. Random approximation (Fig. 5(c)) is not ideal: new congestion probabilities are artificially added mid-year (around summer, days 130–250). Optimized k-medoids (Fig. 5(d)) is very close to the full-year simulation using only 60 representative days (≈ 6x simulation time speedup). See supplementary material for a two-dimensional view of the heatmaps.

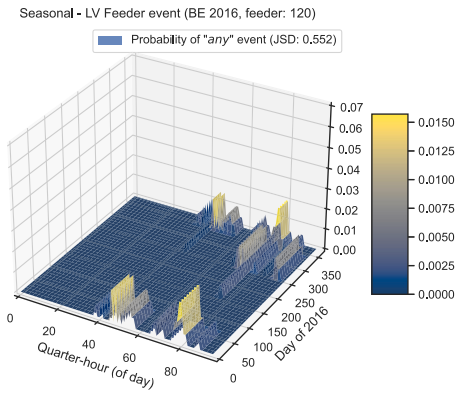
Fig. 6 shows LVG meter representative-day clustering multi-objective performance against benchmarks when we increase the number of clusters k from one up to the maximum of 366. All benchmarks where outperformed by k-medoids. Fig. 7 shows LVG feeder representative-day clustering performance against benchmarks. Again, optimized k-medoids (red line: mean for 146 LVG feeders) outperformed the random clustering (blue line), the static benchmarks (horizontal lines), and the repeat-days clustering (black line). The probability time series that we had at our disposal were generated with power flow simulations, for the "any" congestion event of: over-voltage (OV), over-current (OC) and under-voltage (UV). We present results for all the congestion types together as they showed similar patterns. Worth reminding is that we used the clusters/representative days found from the meters time series and that we intentionally did not use any feeder-related meta-data.

Table 3 shows the LVG feeders congestion event probability divergence of a full year of simulations against the representative-day clustering approach. This table shows how much the quarter-hour congestion event probability (using only 30, 60, 90 or 180 simulated days) differs from a full-year simulation (which requires 366 simulation days for 2016), for all feeders (146 feeder average). Our goal is to achieve a smaller Jensen–Shannon Divergence (JSD) than the "no clusters" case (underlined number: 0.611). By picking (clustering) the most similar days possible and good medoids (cluster centers), divergence becomes smaller. This ablation study shows that our approach can surpass all the benchmarks and random clustering with statistical significance (also for $k = 15$, $k = 20$ and $k = 210$, omitted due to table space constraints). We repeated the experiment 50 times with a different random seed for each run. The "*Baselines*" table section contains the performance of all the benchmarks (no clusters, seasonal clustering, ...). The seasonal and the weekday-based benchmarks (with the suffix "a-week") have some missing values (denoted with a "-") because they have a fixed number of clusters k (e.g. since the one-a-week benchmark has 52 clusters, it is only comparable with $k = 60$ or more). The next table section, "*Mean ± one standard deviation*", presents the mean (average) and the standard deviation of all the 50 experiment runs, for each of the techniques (random clustering, k-medoids, ...). The "*Best*" section shows the experiment with the least error (out of all the 50 repeats). Finally, we used the Welch's t-test [46] of unequal variances to assess statistical significance. It is a two-sample location test to assess the null hypothesis that two populations have unequal means (in our case each technique is compared against random clustering). It is an adaptation of the Student's t-test and it is considered more reliable when the two samples have unequal variances. We provide the t-statistic along with the p-value range denoted with "*" characters.

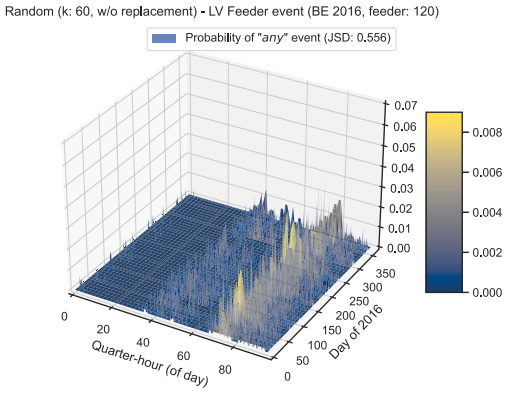
Fig. 8 assesses the LVG meters consumptions clustering balance among: the number of representative days r, reconstruction error reductions against a single representative day (single cluster: 366 samples from all days of 2016, representative day: 1 July 2023, mid-year day 183), and improvements in computation time (speedup in parenthesis). As r increases, all error metrics improve. However, $r = 210$ (63.5% to 100% error decrease) does not seem to offer large enough error reductions against $r = 180$ (56.4% to 99.2% error decrease) while the theoretical speedup (against the full 366 days) improves marginally (2x versus 1.7x). For LVG meters approximations, choosing $r = 15$ (14.6%–72.2% error reduction versus a single day) up to $r = 180$ yields good trade-offs between speed and reconstruction accuracy. For LVG feeder congestion event reconstructions (Fig. 9): the larger the number of representative days r, the larger the error reduction (average error and variance seem to decrease almost linearly). Therefore, we should pick the largest r based on the simulation speedup we want to achieve against a full 366-day simulation. With $r = 15$ (depending on the probabilistic divergence metric: 20%–72% error reduction versus a single representative-day), we achieve a 24.4x simulation speedup. E.g., an LVG feeder simulation (Flanders with approximately 200,000 feeders) of 15 representative days would take ≈5 days of computation,



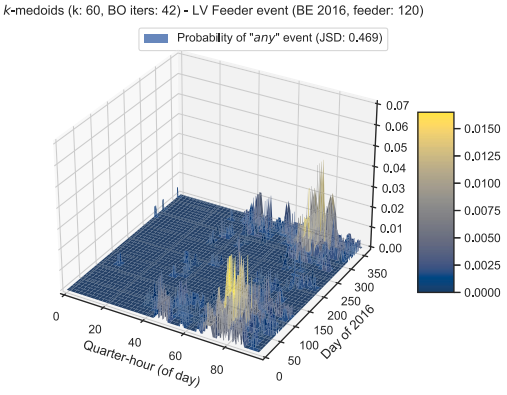
(a) Full-year simulation (366 days)



(b) Seasonal benchmark ($k = 4$)



(c) Random clustering ($k = 60$)



(d) Optimized k-medoids ($k = 60$)

Fig. 5. Reconstruction of feeder event probabilities (3D heatmap from the Belgian (region Flanders) 2016 LVG feeder 120 simulation, first horizontal axis: quarter-hour of a day, second horizontal axis: day of 2016, vertical axis and color: probability of congestion event $\in [0, 1]$). The seasonal clustering (Fig. 5(b)) is too coarse and random clustering (Fig. 5(c)) added erroneous peaks around summer. Lower Jensen–Shannon divergence (JSD) values (0.469 in Fig. 5(d) with 60-day approximation versus 0.552 in Fig. 5(b) with seasonal approximation) indicate a better reconstruction of the full-year (366-day) simulation (Fig. 5(a)) visually. Note: see supplementary material for a two-dimensional view of the heatmaps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

whereas simulating for 366 days (the full year), would take ≈122 days (if one LVG day simulation takes around 8 h of computation). Using $r = 180$ (in other words: simulating half of 2016) we achieve 56%–99% error reduction, with a 2x execution time speedup versus the full year. Note that all error metrics (MAE, MSE, KL and JSD) show the same qualitative scaling as JSD in regards to the number of clusters and representative days.

3.4. Clustering explainability and interpretability

Fig. 10 shows results from the optimized k-medoids ($k = 30$) clustering (Fig. 10(a)), representative days (Fig. 10(b)), average LVG daily consumption (Fig. 10(c)) and daily weather variables (Fig. 10(d) for temperature) for the Flemish LVG in 2016. The lowest daily kWh consumption (Fig. 10(c) with clusters 6, 15, 4: white and light blue days) coincides with the warmest 2016 days (Fig. 10(d) with clusters 4, 16: dark red Jul 19–23, Aug 23–27, Sep 12–15). Seven weekends were clustered distinctly (clusters 9, 3, 6, 16, 24: Jan 16–17, 23–24, Feb 20–21, Apr 23–24, Jul 30–31, Sep 17–18, Dec 17–18) and three Belgian holidays (clusters 27, 28, 1: Feb 14, Sep 24 and Nov 15). Representative days are spread around the year (Fig. 10(b)).

Fig. 11 shows the ranking of the most important exogenous variables of a tree ensemble classifier (trained with stratified cross-validation on the found k-medoids clusters, 94.59% test set accuracy). The six most important were related to photosynthetically active solar radiation, soil water volume, soil temperature, albedo and evaporation. The top three weather variables had the most impact (mean(|SHAP value| >3)): (1) Clear Sky Photosynthetically Active Surface Radiation (PARCS) which focuses on 400–700 nm wavelengths (visible spectrum which coincides with the absorption wavelengths of solar panels), (2) Soil Water Volume (SWVL) at Layer 4 (100–150 cm) and (3) Soil Temperature at Layer 4 (STL). Sea Surface Temperature (SST), Forecast Albedo (FAL) (or in other words: the earth's reflectivity) and Evaporation (E) also had a SHAP value larger than one. Calendar variables like the day/week/month of the year and whether it is a weekend or weekday were also in the top 25 features. Our interpretation of the feature importances is as follows. Different type-of-day features are important as expected. The model finds PARCS more important than Surface short-wave (solar) radiation downwards (SSRD) (which captures the full spectrum of solar radiation), and suggests that the visible spectrum of sunlight is more relevant for residential consumption. This could be related to PV production, since PV mostly absorbs visible sunlight.

Table 3
LVG feeders congestion event probability ("any": over-voltage, over-current and under-voltage events) divergence of a full year versus the representative-day clustering. Data: Belgian Flanders 2016, 146 feeder quarter-hour probabilities. Parenthesis: Welch's t-test of unequal variances [46] t-statistic and p-value statistical significance (against random clustering with *** for p-value < .001). Optimized k-medoids setup: noise-aware multi-objective hyperparameter optimization (22 Sobol + 20 Bayesian iterations), execution times from 28.7 min ± 15.2 min ($k = 15$) up to 37.6 min ± 23.2 min ($k = 180$).

Divergence: "any" feeder congestion event (quarter-hour %, JSD)				
Clustering (50 runs)	Cluster count k			
	30	60	90	180
Baselines				
No clusters	0.611	0.611	0.611	0.611
Seasonal ($k = 4$)	0.552	0.552	0.552	0.552
One-a-week ($k = 53$)	–	0.477	0.477	0.477
Two-a-week ($k = 106$)	–	–	–	0.423
Three-a-week ($k = 158$)	–	–	–	0.375
Four-a-week ($k = 210$)	–	–	–	–
Repeat days	0.503	0.472	0.472	0.42
Mean ± one standard deviation				
Random clustering	0.578 ± 0.007	0.555 ± 0.007	0.527 ± 0.007	0.431 ± 0.010
k-medoids	0.497 (7.47e+01***)	0.481 (6.44e+01***)	0.483 (4.71e+01***)	0.39 (3.71e+01***)
k-medoids (no exogenous)	0.508 (6.47e+01***)	0.495 (5.16e+01***)	0.49 (4.00e+01***)	0.438 (−3.82e+00***)
Noise-aware multi-objective k-medoids	**0.493 ± 0.009** (5.19e+01***)	**0.462 ± 0.010** (5.16e+01***)	**0.427 ± 0.017** (3.91e+01***)	**0.292 ± 0.028** (3.30e+01***)
Best (out of the 50 runs)				
Random clustering	0.565	0.538	0.506	0.412
k-medoids	0.497	0.481	0.483	0.39
k-medoids (no exogenous)	0.508	0.495	0.49	0.438
Noise-aware multi-objective k-medoids	**0.481**	**0.430**	**0.391**	**0.227**

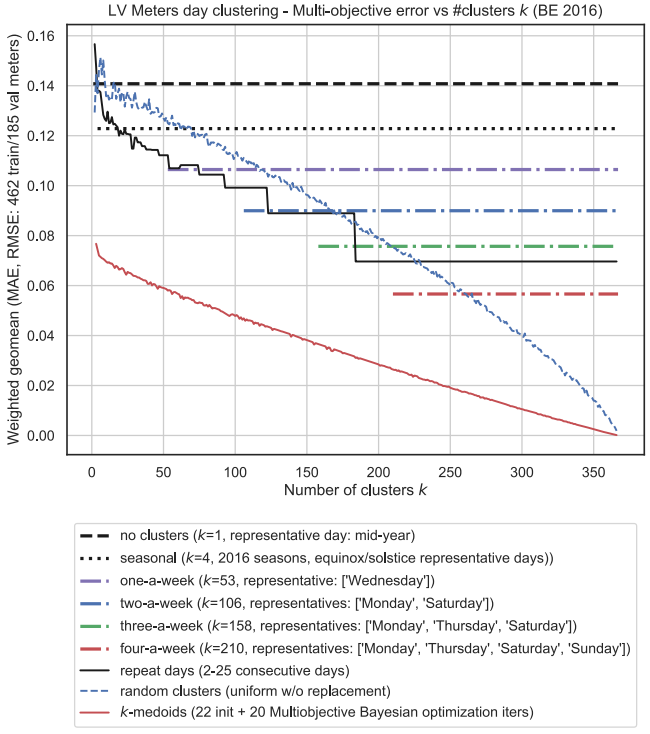


Fig. 6. LVG meter representative-day clustering multi-objective performance against benchmarks when increasing the number of clusters k to the full extent (horizontal axis: number of clusters $k \in [1, 366]$ for 2016, vertical axis: weighted geometric mean of four objectives [MSE, RMSE on the train and validation meters]). The multi-objective Bayesian optimization outperforms all benchmarks. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

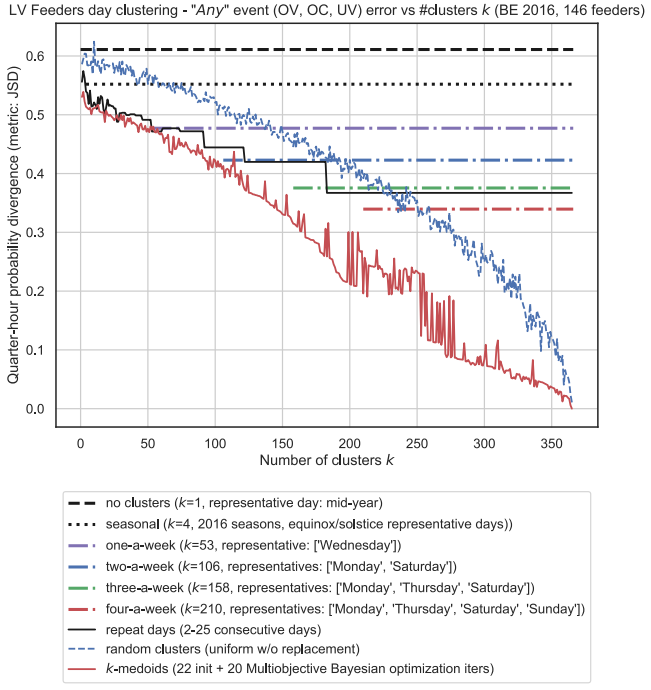


Fig. 7. LVG feeder representative-day clustering performance against benchmarks when increasing the number of clusters k to the full extent (horizontal axis: number of clusters $k \in [1, 366]$ for 2016, vertical axis: average quarter-hour congestion event probability divergence of k-medoids representative days versus the full dataset of 366 days). Again multi-objective Bayesian optimization outperforms all benchmarks. Important to note is that for the clustering we only used LVG meter data and no data related to feeders.

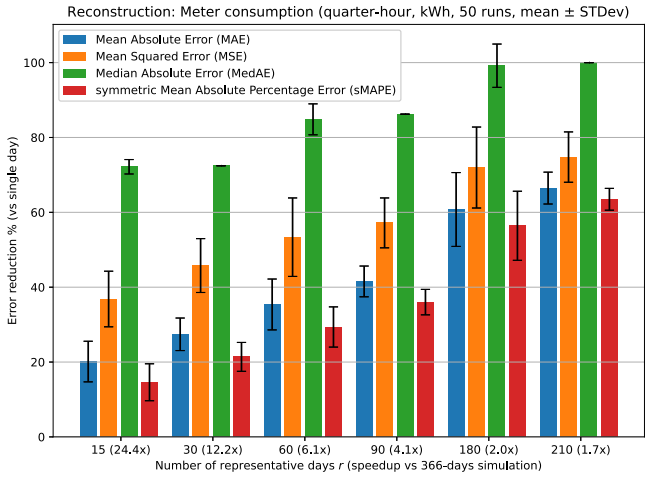


Fig. 8. LVG meters time series reconstruction error: average quarter-hour meter consumption (kWh) versus representative days count. As the number of representative days increases, all metrics (mean and standard deviation of: MAE, MSE, MedAE and sMAPE) show that reconstruction improves. Data: quarter-hour consumption times series of 925 Belgian meters (2016, region Flanders). Note: the single representative day baseline has 366 samples (all days of 2016).

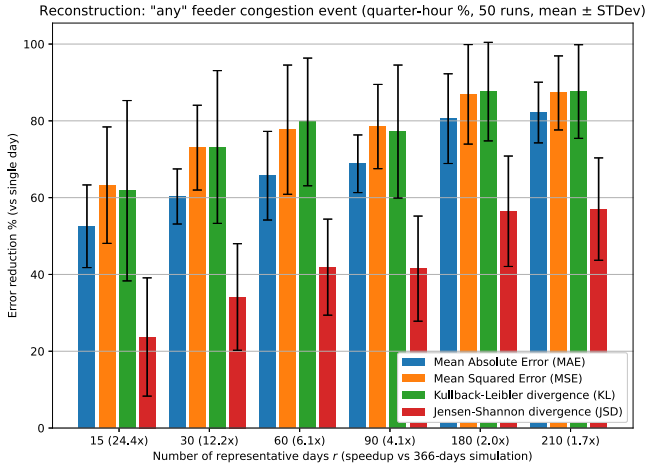


Fig. 9. LVG feeder time series reconstruction error: average quarter-hour congestion event probability (%) versus representative days count. Except for $r = 90$ and $r = 210$: with more representative days, reconstruction accuracy improves (mean and standard deviation of: MAE, MSE, KL and JSD). Data: quarter-hour congestion probability times series of 146 Belgian feeders (2016, region Flanders).

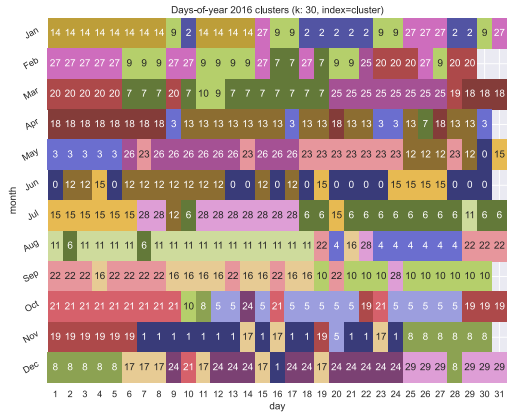
A surprising finding is that the temperature and water content of the soil at high depth (100–150 cm) is found to be important. A possible interpretation is that these high depths are good at capturing memory effects of the weather of previous days. For example, we expect the soil at higher depth to be warm and dry only if the previous days have all been high temperature days and without rain. The fact that surface sea temperature (SST) is also found to be important, corroborates this interpretation. This also aligns with the energy sector insight that the highest consumption on the grids is typically observed at the end of a longer stretch of cold days. A potential hypothesis is that weather measurements with a memory effect are correlated to the behavior of the thermal mass of buildings.

Fig. 12 shows a comparison of the five most important exogenous variables of the most common clusters (optimized k-medoids with $k = 15$): cluster 2 in summer and cluster 26 in winter. If photosynthetically active radiation (first line, PARCS) is high (red values), then it is highly likely that the day belongs to summer cluster 2 (Fig. 12(a): positive SHAP up to 2.8). Otherwise, if PARCS is low (blue values) then it should not belong in cluster 0 of summer (Fig. 12(a): negative SHAP down to −3.2), but in winter cluster 9 (Fig. 12(b): positive SHAP up to 2.9). FAL (albedo) and E (evaporation) show the opposite pattern. SHAP values were calculated after fitting a tree ensemble classifier (94.59% accuracy on test data) with stratified cross-validation on the found k-medoids clusters. Worth noting is that we do not suggest fitting a tree ensemble classifier with more than 90 clusters, because it will be difficult to achieve more than 80% classification accuracy on the test set and there will be very few samples per class (e.g. 365 days/90 clusters ≈ 4 samples per cluster). Fig. 13 shows rule induction (derived from a single decision tree) for choosing the cluster of a day, given exogenous variables. To find the cluster of a day, we start from the top (root) and based on the weather variable value we move left (if the statement is true) or right (if false) until we reach a bottom leaf. In other words, given only the weather and calendar variables of a day, one can deduce in which LVG cluster it should belong. E.g. starting from the root: if we are in the second half of the year (the day of the year, expressed as a sinusoidal wave, is more than −0.009), then we move to the right node (arrow False). Afterwards if the Clear sky surface photosynthetically active radiation is higher than 2077014 J m^{-2} we move right (arrow False), to the forecast albedo node. Now, if the albedo is low, less than 0.155 (low earth reflectivity: black body with high solar radiation absorption, high reflectivity: earth acts like a mirror, reflecting the energy received from the sun), then this day should be long to the summer cluster 2, otherwise to the autumn cluster 5. Most rules contain meteorological variables (like PARCS, SWVL, SST, FAL), but also some calendar ("cos_week", "sin_dayofyear": how far along the week/year expressed as cyclical sinusoidal signals, ...). This tree view provides additional clustering explainability. We can see that several important features from the summary plot (PARCS, FAL, STL4 and calendar-based variables from Fig. 11) do also appear in the decision tree. For this illustration, we fitted a single decision tree classifier (82.43% accuracy on test data), using the found optimized k-medoids indices ($k = 15$) as classes. We used the Tree of Parzen Estimators (TPE) [47] hyperparameter optimization, 51 weather and 12 calendar variables as input train/validation/test data. To avoid class imbalance during training, we randomly and proportionally split the data samples (days of the year) using 10-fold stratified cross-validation.

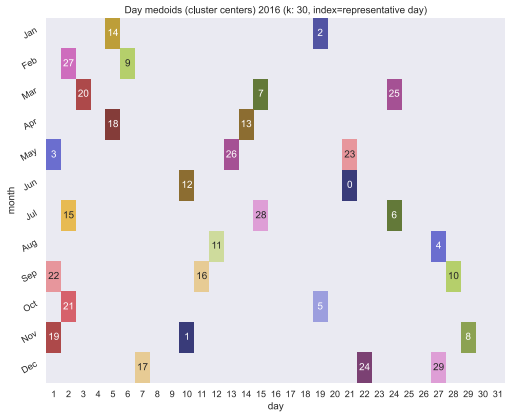
Fig. 14 shows Bayesian optimization hyperparameter statistics. Cosine distance was the most prominent for most k values. PCA was more common for small k (number of clusters) and Kernel PCA for high k. Expit was the most selected exaggerator function and standardization the most common pre-scaler. "build" and "k-medoids++" were the most selected initializers. Finally, the median number of components (matrix decomposition) was around 110 for $k = 15, 180, 210$ and around 50 for $k = 30, 60, 90$. The Bayesian optimizer has also the option to skip some of the steps (see "None" hyperparameter option for the exaggerator and the pre/post-scaler functions in Table 2 and Fig. 14) of the full pipeline (see flowchart in Fig. 4). Having no post-scaler (not shown in figure) was the most common choice. We did not notice distinct patterns for the other hyperparameters (k-medoids method and PCA kernel).

4. Conclusion

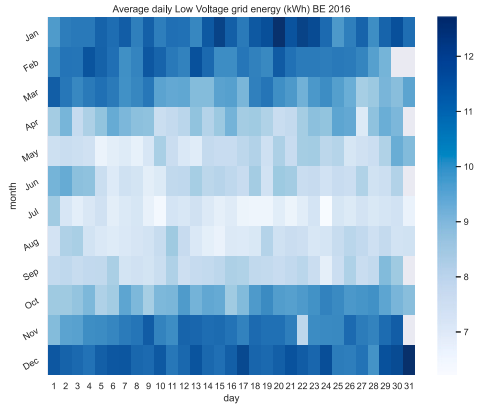
In this paper, we present an end-to-end, statistically robust, scalable and explainable representative-days clustering framework, that can work both at the meter and the feeder level of the LVG. We examined LVG power flow simulation results of congestion event probabilities, along with real-world LVG meter data (from Flanders, Belgium). Handling missing values, extreme outliers and noise are integral parts of the framework. Via dimensionality reduction, clustering remains



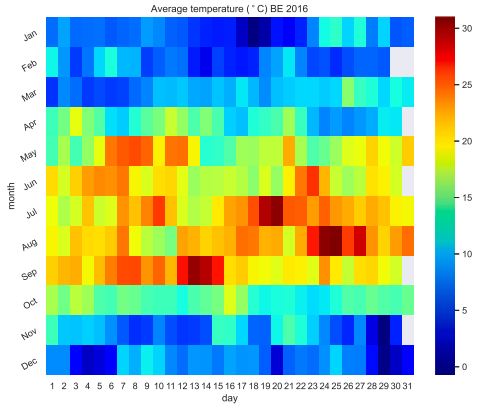
(a) Clustered days (30 clusters)



(b) Representative days (30 medoids)



(c) Average daily LVG consumption



(d) Daily average temperature (2m above surface) (2T)

Fig. 10. Results from MOBO qNEHVI optimized k-medoids ($k = 30$) clustering (Fig. 10(a)), representative days (Fig. 10(b)), average daily LVG consumption (Fig. 10(c)) and daily weather variables (Fig. 10(d) illustrates the average daily temperature) for the Flemish LVG in 2016 (color heatmap horizontal axis: day of month $\in [1, 31]$, vertical axis: month (January to December) of 2016, color: cluster index or variable intensity). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

highly scalable, regardless of the amount of meters and the sampling frequency (we utilized a large dataset of 937 m with quarter-hour sampling frequency, along with 51 daily meteorological features and 12 calendar cycles/event variables). Noise-aware multi-objective Bayesian hyperparameter optimization optimizes the representative-days clustering at the finest level of time granularity (quarter-hour). To validate the clustering performance, we performed an ablation study of 50 random runs with statistical significance tests and introduced several new representative-day benchmarks. Finally, by reformulating the clustering problem into a classification one, we can provide meteorological interpretability insights and increased model explainability via ranked feature importance and decision rule induction. Important to note is that, extreme-outlier removal should be treated as an optional component by the modeler, as it could potentially remove inliers (values that lie within the data distribution and are not erroneous).

Using four time series metrics, our findings indicate that: from 15 up to 180 representative days can be used, for both LVG meters consumption and LVG feeder congestion event approximations. Random representative-day selection showed erroneous artifacts in the summer months (there should not be a high probability of congestion events in those months). The representative days (cluster centers) were spread around the year. Most predictive meteorological variables (for the examined 2016 dataset, using: visual heatmap inspection, ranked feature importance and decision rule induction): surface photosynthetically active radiation (≈400–700 nm wavelength, solar panel absorption wavelengths), soil water volume, soil temperature (with summer clusters having the inverse response compared to the winter ones) and albedo. In general, weather variables that capture climate history were the most impactful in the 2016 LVG event clustering. Calendar features (like weekends, day of the year, ...) are important in choosing day-clusters.

Finally, regarding limitations, there are three aspects: (1) the proposed framework is not tailored to the specific metadata for each feeder, (2) implementing the full end-to-end framework is complex and (3) the framework is more computationally heavy compared to the off-the-shelf k-medoids clustering.

For future work, we would like to: (1) examine additional clustering distance metrics (frequency and amplitude-based), (2) compare the feature importance against different (and more recent) years and (3) contrast the feature importance between different types/subsets of meters and feeders (meters with or without a heat pump, solar panel installations, electrical vehicles, industrial versus residential regions, ...).

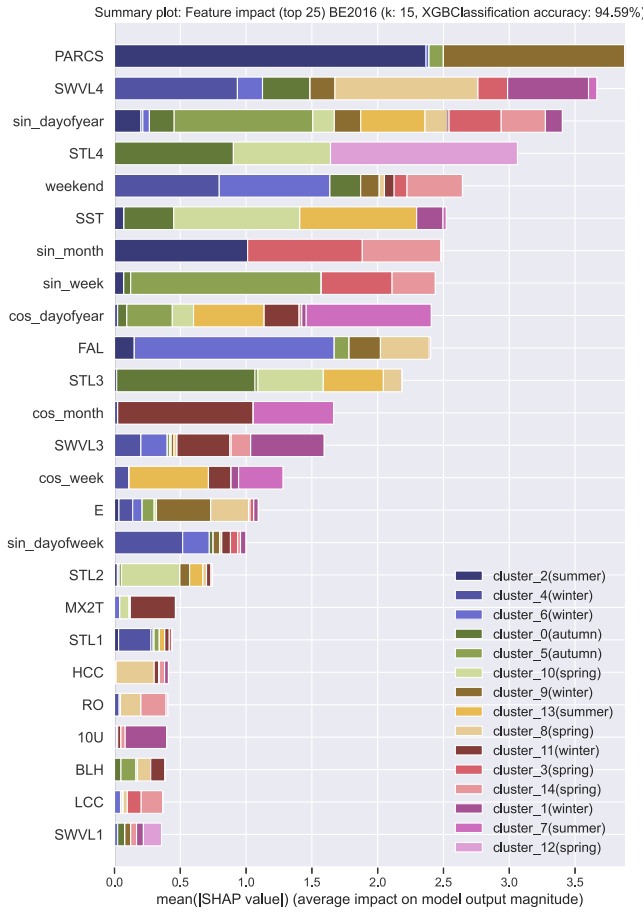
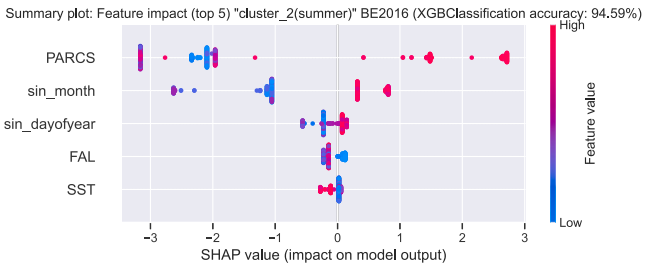
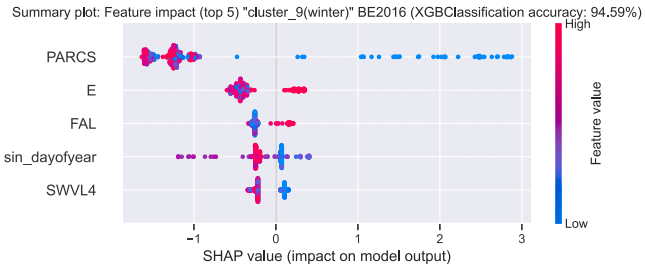


Fig. 11. Ranking of the 25 most important exogenous variables (vertical axis: variable, horizontal axis: average absolute SHAP value) after fitting a tree ensemble classifier with stratified cross-validation on the found k-medoids clusters (with fixed $k = 15$). Top six: (1) Clear Sky Photosynthetically Active Surface Radiation (PARCS) (400–700 nm visible spectrum, solar panel absorption wavelengths), (2) Soil Water Volume (SWVL), (3) Soil Temperature (STL), (4) Sea Surface Temperature (SST), (5) Albedo (FAL) and (6) Evaporation (E). This figure also illustrates that different cluster indices correlate with different combinations of variables (e.g. cluster 2 summer in dark blue color mostly depends on PARCS, whereas the winter cluster 4 in light blue depends a lot on FAL which is 10th overall). Hyperparameters fitted with Tree of Parzen Estimators (TPE) [47]. Note: see supplementary material for descriptions of all 51 meteorological and 12 calendar variables. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Feature importance for cluster 2 in summer



(b) Feature importance for cluster 9 in winter

Fig. 12. Comparison of the five most important exogenous variables (vertical axis: variable, horizontal axis: SHAP value, $k = 15$) for summer cluster 2 (Fig. 12(a)) and winter cluster 9 (Fig. 12(b)). If photosynthetically active radiation (PARCS, 400–700 nm wavelength) is high (red values), the day should belong to summer cluster 2, otherwise (blue values) to winter cluster 9. Albedo (FAL) and evaporation (E) show the opposite pattern. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

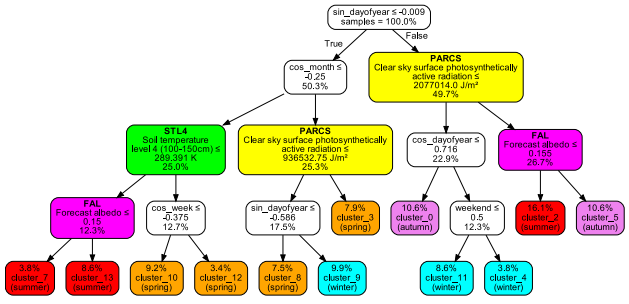


Fig. 13. Rule induction for choosing the cluster of a day, given only exogenous variables (illustrative example with 15 clusters). This decision tree offers additional clustering explainability by extracting rules (if-then-else statements), given the input meteorological features and the output LVG clusters. Classification decision tree accuracy: 82.43% on the test set (with the 11 most-prominent clusters, hyperparameters fitted with Tree of Parzen Estimators (TPE) [47]).

CRediT authorship contribution statement

Konstantinos Theodorakos: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Oscar Mauricio Agudelo:** Writing – review & editing, Supervision, Project administration, Methodology, Conceptualization. **Thijs Becker:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Investigation, Data curation, Conceptualization. **Koen Vanthournout:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Conceptualization. **Reinhilde D'hulst:** Writing – review & editing, Supervision, Conceptualization. **Bart De Moor:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the Flemish DSO Fluvius for supporting this work and for providing the consumption data. This work is supported by : • KU Leuven: Research Fund (projects iBOF/23/064, C3/20/117, C3I-21-00316), Industrial Research Fund and several Leuven Research and Development bilateral industrial projects; • Flemish Government Agencies: ∘ FWO: SBO project S005319N, TBM Project T001919N; FWO PhD Grant 11K5623N; ∘ EWI: the Flanders AI Research Program ∘ VLAIO: CSBO (HBC.2021.0076) Baekeland PhD (HBC.20192204) • European Commission: European Research Council under the European Union's Horizon 2020 research and innovation programme (ERC Adv. Grant grant agreement No 885682).

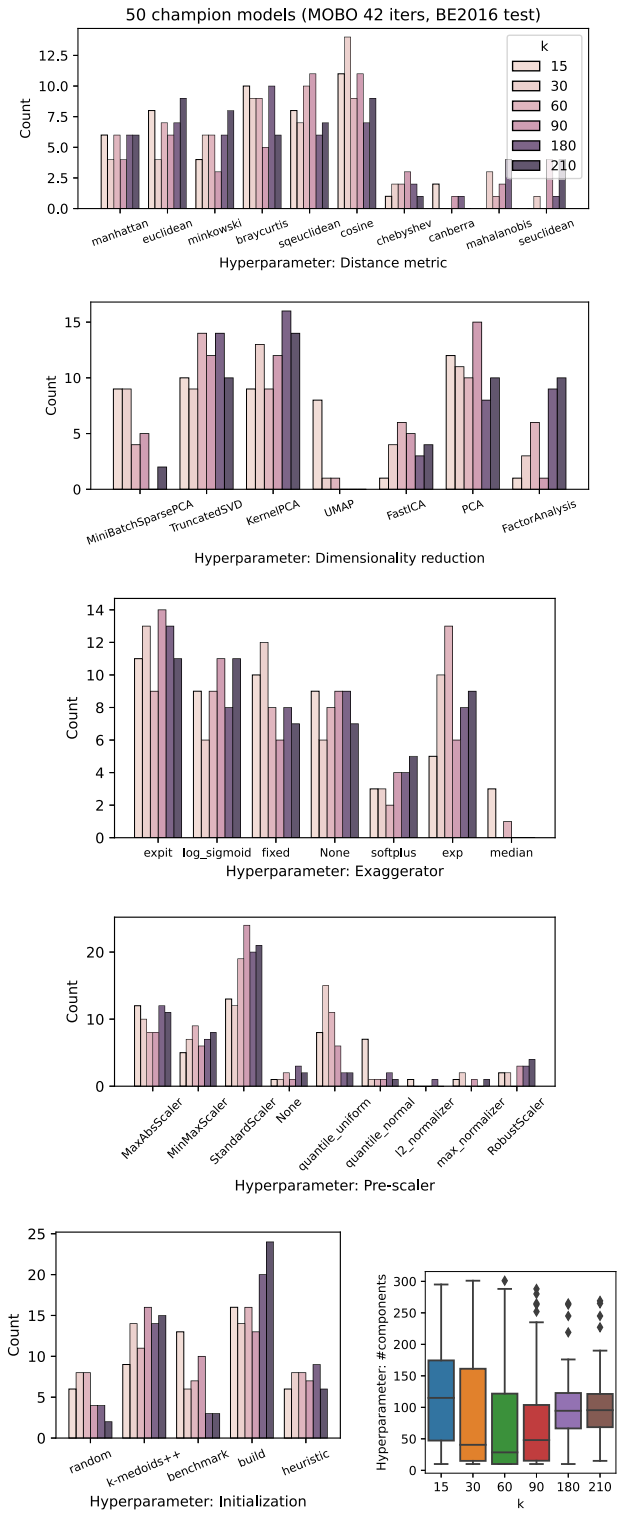


Fig. 14. Histograms of the Bayesian optimization hyperparameters (from the top 50 champion k-medoids models). From top to bottom: Distance metric, matrix decomposition methods for dimensionality reduction, exaggerator function (that squeezes and expands raw values), data pre-scaler, k-medoids clustering initializer and number of decomposition components ($\in [10, 300]$).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.segan.2025.101622.

Data availability

The authors do not have permission to share data.

References

[1] S. Elmallah, A.M. Brockway, D. Callaway, Can distribution grid infrastructure accommodate residential electrification and electric vehicle adoption in Northern California? Environ. Res.: Infrastruct. Sustain. 2 (4) (2022) 45005, http://dx.doi.org/10.1088/2634-4505/ac949c.

[2] E. Proedrou, A comprehensive review of residential electricity load profile models, IEEE Access 9 (2021) 12114–12133, http://dx.doi.org/10.1109/ACCESS.2021.3050074.

[3] T. Unterluggauer, F. Hipolito, J. Rich, M. Marinelli, P.B. Andersen, Impact of cost-based smart electric vehicle charging on urban low voltage power distribution networks, Sustain. Energy Grids Netw. 35 (2023) 101085, http://dx.doi.org/10.1016/j.segan.2023.101085.

[4] P. Hogeveen, M. Steinbuch, G.P.J. Verbong, A. Wargers, Revisiting static charge schedules for electric vehicles as temporary solution to low-voltage grid congestion with recent charging and grid data, Sustain. Energy Grids Netw. 31 (2022) 100701, http://dx.doi.org/10.1016/j.segan.2022.100701.

[5] A.F. Cortés Borray, J. Merino, E. Torres, A. Garcés, J. Mazón, Centralised coordination of EVs charging and PV active power curtailment over multiple aggregators in low voltage networks, Sustain. Energy Grids Netw. 27 (2021) 100470, http://dx.doi.org/10.1016/j.segan.2021.100470.

[6] J. Soenen, A. Yurtman, T. Becker, R. D'hulst, K. Vanthournout, W. Meert, H. Blockeel, Scenario generation of residential electricity consumption through sampling of historical data, Sustain. Energy Grids Netw. 34 (2023) 100985, http://dx.doi.org/10.1016/j.segan.2022.100985.

[7] J. Soenen, L. Botman, K. Theodorakos, D.V. Daele, A. Yurtman, J. Bekker, K. Vanthournout, Enabling the energy transition with smart meter data, 2021, URL https://ai.kuleuven.be/stories/post/2021-11-03-energy-forecasting/.

[8] G. Giasemidis, S. Haben, Modelling the demand and uncertainty of low voltage networks and the effect of non-domestic consumers, Sustain. Energy Grids Netw. 16 (2018) 341–350, http://dx.doi.org/10.1016/j.segan.2018.10.002.

[9] K. Poncelet, H. Höschle, E. Delarue, A. Virag, W. D'haeseleer, Selecting representative days for capturing the implications of integrating intermittent renewables in generation expansion planning problems, IEEE Trans. Power Syst. 32 (3) (2017) 1936–1948, http://dx.doi.org/10.1109/TPWRS.2016.2596803.

[10] F.J.D. Sisternes, M.D. Webster, O.J.D. Sisternes, M.D. Webster, Optimal Selection of Sample Weeks for Approximating the Net Load in Generation Planning Problems, ESD Working Papers, Massachusetts Institute of Technology. Engineering Systems Division, 2013.

[11] X. Wang, Z. Yao, M. Papaefthymiou, A real-time electrical load forecasting and unsupervised anomaly detection framework, Appl. Energy 330 (2023) 120279, http://dx.doi.org/10.1016/J.APENERGY.2022.120279.

[12] Y. Liu, R. Sioshansi, A.J. Conejo, Hierarchical clustering to find representative operating periods for capacity-expansion modeling, IEEE Trans. Power Syst. 33 (3) (2018) 3029–3039, http://dx.doi.org/10.1109/TPWRS.2017.2746379.

[13] C. Li, A.J. Conejo, J.D. Siirola, I.E. Grossmann, On representative day selection for capacity expansion planning of power systems under extreme operating conditions, Int. J. Electr. Power Energy Syst. 137 (2022) 107697, http://dx.doi.org/10.1016/j.ijepes.2021.107697.

[14] I.J. Scott, P.M.S. Carvalho, A. Botterud, C.A. Silva, Clustering representative days for power systems generation expansion planning: Capturing the effects of variable renewables and energy storage, Appl. Energy 253 (2019) 113603, http://dx.doi.org/10.1016/j.apenergy.2019.113603.

[15] A. García-Cerezo, L. Baringo, R. García-Bertrand, Representative days for expansion decisions in power systems, Energies 13 (2) (2020) http://dx.doi.org/10.3390/en13020335.

[16] G.W. Hart, Nonintrusive appliance load monitoring, Proc. IEEE 80 (12) (1992) 1870–1891, http://dx.doi.org/10.1109/5.192069.

[17] Y. Himeur, A. Alsalemi, F. Bensaali, A. Amira, Robust event-based non-intrusive appliance recognition using multi-scale wavelet packet tree and ensemble bagging tree, Appl. Energy 267 (2020) 114877, http://dx.doi.org/10.1016/j.apenergy.2020.114877.

[18] L. Yan, W. Tian, H. Wang, X. Hao, Z. Li, Robust event detection for residential load disaggregation, Appl. Energy 331 (2023) 120339, http://dx.doi.org/10.1016/j.apenergy.2022.120339.

[19] M. Sun, F. Teng, X. Zhang, G. Strbac, D. Pudjianto, Data-driven representative day selection for investment decisions: A cost-oriented approach, IEEE Trans. Power Syst. 34 (4) (2019) 2925–2936, http://dx.doi.org/10.1109/TPWRS.2019.2892619.

[20] K. Fahy, M. Stadler, Z.K. Pecenak, J. Kleissl, Input data reduction for microgrid sizing and energy cost modeling: Representative days and demand charges, J. Renew. Sustain. Energy 11 (6) (2019) 065301, http://dx.doi.org/10.1063/1.5121319.

[21] H. Teichgraeber, A.R. Brandt, Clustering methods to find representative periods for the optimization of energy systems: An initial framework and comparison, Appl. Energy 239 (2019) 1283–1293, http://dx.doi.org/10.1016/j.apenergy.2019.02.012.

[22] L.L. Raket, B. Grimme, G. Schöner, C. Igel, B. Markussen, Separating timing, movement conditions and individual differences in the analysis of human movement, PLoS Comput. Biol. 12 (9) (2016) 1–27, http://dx.doi.org/10.1371/journal.pcbi.1005092.

[23] E. Vidal, F. Casacuberta, J.M. Benedi, M.J. Lloret, H. Rulot, On the verification of triangle inequality by dynamic time-warping dissimilarity measures, Speech Commun. 7 (1) (1988) 67–79, http://dx.doi.org/10.1016/0167-6393(88)90022-2.

[24] Y. Wang, Q. Chen, T. Hong, C. Kang, Review of smart meter data analytics: Applications, methodologies, and challenges, IEEE Trans. Smart Grid 10 (3) (2019) 3125–3148, http://dx.doi.org/10.1109/TSG.2018.2818167.

[25] S. Daulton, M. Balandat, E. Bakshy, Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement, in: M.R. Vaughan, A. Beygelzimer, Y. Dauphin, P. Liang, J. Wortman (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc., 2021, pp. 2187–2200.

[26] D.P. Dee, S.M. Uppala, A.J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M.A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A.C. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A.J. Geer, L. Haimberger, S.B. Healy, H. Hersbach, E.V. Hólm, L. Isaksen, P. Kållberg, M. Köhler, M. Matricardi, A.P. Mcnally, B.M. Monge-Sanz, J.J. Morcrette, B.K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J.N. Thépaut, F. Vitart, The ERA-interim reanalysis: Configuration and performance of the data assimilation system, Q. J. R. Meteorol. Soc. 137 (656) (2011) 553–597, http://dx.doi.org/10.1002/qj.828.

[27] P.J. Rousseeuw, C. Croux, Alternatives to the median absolute deviation, J. Amer. Statist. Assoc. 88 (424) (1993) 1273–1283, http://dx.doi.org/10.1080/01621459.1993.10476408.

[28] C. De Boor, A Practical Guide to Splines, Revised, Springer, 2001, pp. 207–214.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in {P}ython, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[30] N. Halko, P.G. Martinsson, J.A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. 53 (2) (2011) 217–288, http://dx.doi.org/10.1137/090771806.

[31] B. Schölkopf, A.J. Smola, K.-R. Müller, Kernel principal component analysis, in: Advances in Kernel Methods: Support Vector Learning, The MIT Press, 1998, http://dx.doi.org/10.7551/mitpress/1130.003.0026.

[32] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications., Neural Netw. 13 (4) (2000) 411–430, http://dx.doi.org/10.1016/s0893-6080(00)00026-5.

[33] L. McInnes, J. Healy, N. Saul, L. Großberger, UMAP: Uniform manifold approximation and projection, J. Open Sour. Softw. 3 (29) (2018) 861, http://dx.doi.org/10.21105/joss.00861.

[34] D. Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012, p. 468.

[35] J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian optimization of machine learning algorithms, in: F.P. Weinberger, C. Burges, L. Bottou, K.Q. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc., Red Hook, NY, USA, 2012.

[36] K. Theodorakos, O. Mauricio Agudelo, J. Schreurs, J.A.K. Suykens, B. De Moor, Island transpeciation: A co-evolutionary neural architecture search, applied to country-scale air-quality forecasting, IEEE Trans. Evol. Comput. 27 (4) (2023) 878–892, http://dx.doi.org/10.1109/TEVC.2022.3189500.

[37] S. Daulton, M. Balandat, E. Bakshy, Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, Curran Associates Inc., Vancouver, BC, Canada, 2020, p. 14, http://dx.doi.org/10.5555/3495724.3496550.

[38] H. Niederreiter, Low-discrepancy and low-dispersion sequences, J. Number Theory 30 (1) (1988) 51–70, http://dx.doi.org/10.1016/0022-314X(88)90025-X.

[39] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental algorithms for scientific computing in python, Nature Methods 17 (2020) 261–272, http://dx.doi.org/10.1038/s41592-019-0686-2.

[40] J. Grossman, M. Grossman, R. Katz, Averages: A New Approach, Archimedes Foundation, 1983.

[41] J.M. Joyce, Kullback-Leibler divergence, in: M. Lovric (Ed.), International Encyclopedia of Statistical Science, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 720–722, http://dx.doi.org/10.1007/978-3-642-04898-2{_}327.

[42] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86, http://dx.doi.org/10.1214/aoms/1177729694.

[43] F. Nielsen, On the Jensen–Shannon symmetrization of distances relying on abstract means, Entropy 21 (5) (2019) http://dx.doi.org/10.3390/e21050485.

[44] S.M. Lundberg, G. Erion, H. Chen, A. DeGrave, J.M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S.-I. Lee, From local explanations to global understanding with explainable AI for trees, Nat. Mach. Intell. 2 (1) (2020) 56–67, http://dx.doi.org/10.1038/s42256-019-0138-9.

[45] A.E. Ezugwu, A.M. Ikotun, O.O. Oyelade, L. Abualigah, J.O. Agushaka, C.I. Eke, A.A. Akinyelu, A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects, Eng. Appl. Artif. Intell. 110 (2022) 104743, http://dx.doi.org/10.1016/j.engappai.2022.104743.

[46] B.L. Welch, The generalization of ‘student’s’ problem when several different population variances are involved, Biometrika 34 (1–2) (1947) 28–35, http://dx.doi.org/10.1093/biomet/34.1-2.28.

[47] J. Bergstra, D. Yamins, D. Cox, Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures, in: S. Dasgupta, D. McAllester (Eds.), Proceedings of the 30th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 28, (1) PMLR, Atlanta, Georgia, USA, 2013, pp. 115–123.