Online Adaptive Clustering Algorithm for Load Profiling

G. Le Ray*, P. Pinson

Centre for Electric Power and Energy, Technical University of Denmark, Kgs. Lyngby, Denmark

Abstract

With the large scale deployment of smart metering, energy sector is facing 'Big Data' related challenges. While metered customers generate streams of data, load profiling methods are not taking advantage of this structure. Indeed, insights on the demand are traditionally provided by static typical load profiles. Renewable energy sources generate intermittency in the production and subsequently uncertainty in aligning the generation to the demand at any time. This work proposes a new view on load profiling that takes benefit of the stream structure of the data, an adaptive and recursive clustering method that generates typical load profiles updated to newly collected data. The online adaptive clustering algorithm is based on an online K-means approach using a dynamic time warping based distance associated with a facility location to adjust the number of typical load profiles. The performance of the algorithm is evaluated on a synthetic dataset and applications are presented on real world dataset from both electricity and central district heating.

*Corresponding author

Email address: gleray@elektro.dtu.dk (G. Le Ray)

Keywords: Clustering, load profiling, smart grid, time-series analysis.

1. Introduction

The energy sector is following the trends of Big Data and growing interest is placed in collecting and analyzing energy data. The development of Advanced Metering Infrastructure (AMI), and Information and Communication Technologies (ICT), thanks to public investments (Third Energy Package in Europe [1] and the American Recovery and Reinvestment Act in the United States of America [2]), constitutes the informational backbone of the energy sector. These investments were motivated by the implementation of 'greener' energy policies supporting the use of larger amount of Renewable Energy Sources (RES). RES are known to introduce intermittency in the production, hence, if no actions are taken, the larger the share of RES in the generation, the farther it can be from the consumption. It is then a necessity to have information about the demand status at a resolution that allows operators to take actions to minimize the use of conventional energy sources.

Typical load profiles, that describe the consumption of group of customers, are used to provide information to the utilities about the demand status. Load profiles have been mainly used in the electricity sector, but other energy fields (e.g. central district heating, gas) can benefit from its development. The Electricity sector is also the main leverage in transitioning to more sustainable energy production/consumption providing the highest share of RES. As production has to meet the consumption at any time, RES intermittency has to be compensated on the consumption side. Solutions

exist to optimize the use of RES (i.e. storage, demand side management) but they all rely on precise information on generation and demand statuses.

Typical load profiles were traditionally generated for the electricity sector by segmenting customers by activities. With only a few customers metered, demographics or customers categories available, the goal was to estimate load profiles of non-metered customers, using assumptions to extrapolate hourly load profiles from yearly consumption. It was known to be inaccurate as sub-populations exist in almost each category [3]. Deployment of AMI has drastically changed the paradigm of load profiling from estimating the demand using a few metered customers to summarizing the information contained in a large pool of metered customers.

State-of-the-art load profiling methods use clustering algorithms combined with dimension reduction techniques on batches of historical data and generate static typical load profiles which suppose that loads are repeated over years taking into account seasonality and other temporal periodicity of the data [4, 3, 5, 6, 7]. This is a first step toward data-driven load profiling. Nevertheless, technological evolution of white appliances and the penetration of new type of electrical appliances (e.g. electric vehicle, heat pumps, batteries) are actually changing the loads on a time scale which requires the profiling to be rerun more often. A recursive approach to load profiling would be a computationally inexpensive way to do so. The problem of temporal dependence, inherent to time series, has been tackled by using Wavelet transform [8] or Fourier transform [9] that can be seen as dimension reduction techniques. However, both the number of clusters and the assignment of customers to a cluster remain static and the clustering has

to be rerun to be updated with newly collected data. Bayesian framework is by essence inferential and is a solution to the problem of updating the clusters to newly collected data. Example of Gaussian mixture models [10] and Dirichlet process [11] can be found in the literature but none of them challenges the temporal static structure of the clustering.

Table 1: The different type of clustering-based load profiling [12].

number of clusters:	static	dynamic
load static	Type 1	Type 2
load dynamic	Type 3	Type 4

Benítez *et al.* have defined four types of clustering-based load profiling (Table 1) depending on whether loads are considered dynamic or static and whether the number of clusters K is evolving or not. In this categorization, the work in [5, 6, 7, 8, 9, 10, 11] are Type 1. In their latest works, Benítez *et al.* have implemented type 2 [12] and type 3 [13] clustering-based load profiling. In this paper, we present a Type 4 clustering-based load profiling methodology. The clustering process is i) flexible, customers can change cluster ; ii) adaptive, the number of clusters can change according to data structure; iii) online, the typical load profiles are dynamic and recursively updated and iv) it respects time dependency of load patterns.

The remainder of this paper is organized as follow: Section 2 presents the notations and the preprocessing of the data, the methodology is introduced in Section 3, the performance of the online adaptive clustering is evaluated on a synthetic data set in Section 4. Real-world data applications are presented in Section 5 and the work is ultimately concluded with an outlook in Section 6.

2. Preliminaries: Notations and Data Preprocessing

The notations used to present the online adaptive clustering algorithm in the following sections are first introduced. Smart meters can record consumption at high frequency (up to 1 second), however data are broadcast into batches at regular intervals during a day to minimize the communication costs. In this framework, metering data are then not collected online but as blocks of data $\mathbf{\Omega}^t$ of fixed length (e.g. days, weeks). Each block $\mathbf{\Omega}^t$, $t \in \{0, \ldots, T\}$ consists into a set of vectors

$$\mathbf{\Omega}^t = \{\mathbf{X}_1^t, \ldots, \mathbf{X}_i^t, \ldots, \mathbf{X}_I^t\}, \tag{1}$$

where each vector $\mathbf{X}_i^t$ is a load from meter i at time step t with all the same length (e.g. 24 for blocks of a day with hourly resolution). The set of typical load profiles $\mathbf{\Upsilon}^t$ generated after clustering $\mathbf{\Omega}^t$ at step t is formed by K^t vectors

$$\mathbf{\Upsilon}^t = \{\mathbf{Y}_1^t, \ldots, \mathbf{Y}_k^t, \ldots, \mathbf{Y}_{K^t}^t\}, \tag{2}$$

where K^t is the number of clusters at time step t. The clustering algorithm used in this work is distance-based. The distances between the set of loads $\mathbf{\Omega}^t$ and the typical load profiles $\mathbf{\Upsilon}^{t-1}$ at previous time step $t-1$ are calculated and stored into a matrix

$$\mathbf{D}^t = d(\mathbf{\Omega}^t, \mathbf{\Upsilon}^{t-1}) = \begin{bmatrix} d_{11}^t & \ldots & d_{1k}^t & \ldots & d_{1K^t}^t \\ \vdots & \ddots & & & \vdots \\ d_{i1}^t & & d_{ik}^t & & d_{iK^t}^t \\ \vdots & & & \ddots & \vdots \\ d_{I1}^t & \ldots & d_{Ik}^t & \ldots & d_{IK^t}^t \end{bmatrix}. \tag{3}$$

The distance between a load $\mathbf{X}_i^t$ and a typical load profile $\mathbf{Y}_k^{t-1}$ is noted d_{ik}^t. A vector of labels $\mathbf{A}^t = [A_1^t, \ldots, A_I^t]$ specifying to which typical load profile k each individual load $\mathbf{X}_i^t$ in $\mathbf{\Omega}^t$ is assigned to is generated using the operator,

$$\forall i \in [1, I], A_i^t = \underset{k}{\operatorname{argmin}}(\mathbf{d}_i^t) \tag{4}$$

on each line $\mathbf{d}_i^t$ of matrix $\mathbf{D}^t$. The result of assigning each load to a typical load profile is a partition of the loads $\mathbf{\Pi}^t = \{\mathbf{P}_1^t, \ldots, \mathbf{P}_k^t, \ldots, \mathbf{P}_{K^t}^t\}$. As an example, we define a set of five loads $\{\mathbf{X}_1, \ldots, \mathbf{X}_5\}$, and a possible partition into two clusters could be, $\mathbf{\Pi} = \{\mathbf{P}_1 = \{\mathbf{X}_1, \mathbf{X}_3, \mathbf{X}_4\}, \mathbf{P}_2 = \{\mathbf{X}_2, \mathbf{X}_5\}\}$.

If raw loads are clustered, the main information used to cluster would be the average consumption, the amplitude of the peak and the exact peak location. To make the loads comparable, the loads $\mathbf{X}_i^t$ have to be normalized. In this work, we have opted to divide each load by their reference power (i.e. maximum consumption over the period) so that they are bounded to $[0, 1]$ [5].

In the remainder of this paper, the t index will be omitted to simplify the notation when objects from the same time step are used.

3. Proposal Clustering Algorithm

The splitting of this section into subsections is materialized by dotted rectangles in Figure 1. Section 3.1 explains the setting of the online clustering algorithm parameters. Section 3.2 presents the online clustering that consists into an iterative process based on the K-means algorithm that connects time steps. Section 3.3 presents the facility location approach, used to evaluate if extra cluster centers should be created.

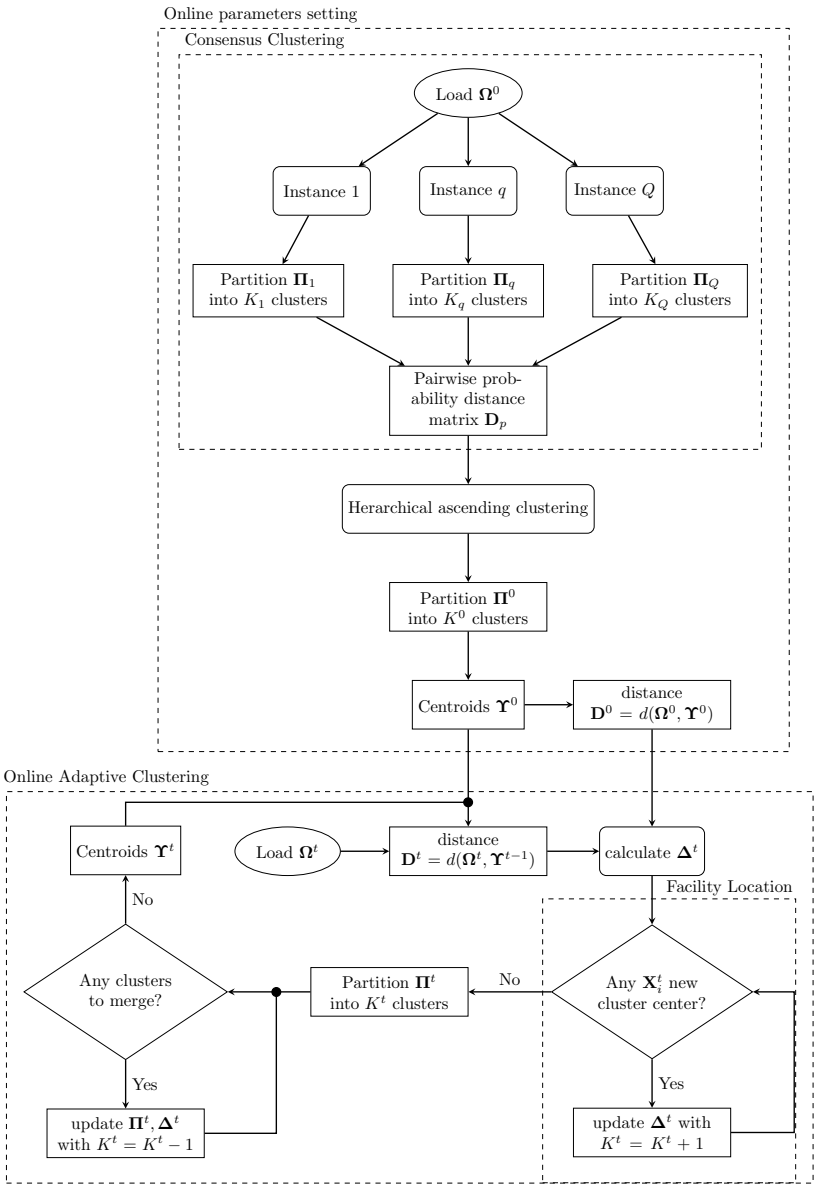


Figure 1: Overview of the online adaptive clustering algorithm.

3.1. Consensus Clustering: Online parameters setting

The online adaptive clustering algorithm requires three parameters: $\boldsymbol{\Upsilon}^0$, the first set of typical load profiles; K^0, the number of cluster centers at $t = 0$ and $d(\boldsymbol{\Omega}^0, \boldsymbol{\Upsilon}^0)$, the distance matrix at $t = 0$. To set these parameters, available historical data can be used offline to create the first partition $\boldsymbol{\Pi}^0$.

Determining the optimal number of clusters K without *prior* information is a non-trivial and computationally expensive problem. The classical way to determine it is by running several instances, $q \in \{1, \ldots, q, \ldots, Q\}$, of a clustering algorithm with different values of K_q to select K^0 that minimizes a criterion [14]. Thereafter only the partition generated with the optimal K^0 is kept, the other instances being used only to determine K^0. In this work, we opted for consensus (or ensemble) clustering to determine K^0 and generate a robust partition [15]. Consensus clustering consists into running in parallel several instances q of clustering algorithm(s) (possibly different) with different values of K_q that return q partitions $\boldsymbol{\Pi}_q$ of historical data $\boldsymbol{\Omega}^0$. Every instance contributes to determine K^0, the partition $\boldsymbol{\Pi}^0$ and the cluster centroids $\boldsymbol{\Upsilon}^0$. From the partitions $\{\boldsymbol{\Pi}_1, \ldots, \boldsymbol{\Pi}_q, \ldots, \boldsymbol{\Pi}_Q\}$ a probability distance $\mathbf{D}_p$ is calculated,

$$\mathbf{D}_p(i, l) = 1 - \frac{\sum_{q=1}^{Q} K_q \delta(\mathbf{A}_{qi}, \mathbf{A}_{ql})}{\sum_{q=1}^{Q} K_q}, \tag{5}$$

$$\text{with } \delta(a, b) = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{if } a \neq b \end{cases} \tag{6}$$

δ the co-occurrence matrix specifies if two points are in the same cluster in each instance and creates a linkage between each pair of loads in $\boldsymbol{\Omega}^0$. The sum of the co-occurrence matrices is weighted by the number of clusters K_q

set in each instance. The benefits of consensus clustering are multiple; all instances are used to build the probability distance matrix; the final partition is a consensus between the different instances so it reduces the bias of each instance; it gets freed from a potential bias due to the choice of initial set of cluster centers.

Using the dendrogram of the hierarchical ascending clustering implemented on the probability distance matrix with the Ward criterion, the user determines the number of clusters K^0 . The expertise of the user in solving a specific problem is necessary to determine the optimal number of clusters K^0. From the K^0 clusters, the initial typical load profiles, $\boldsymbol{\Upsilon}^0$ are calculated by the average of the loads in each cluster, and the initial distance matrix $\mathbf{D}^0 = d(\boldsymbol{\Omega}^0, \boldsymbol{\Upsilon}^0)$ of each loads in $\boldsymbol{\Omega}^0$ to the typical load profiles $\boldsymbol{\Upsilon}^0$ is generated.

3.2. Online Clustering

The online clustering algorithm takes as input the results of the consensus clustering, the typical load profiles $\boldsymbol{\Upsilon}^0$ and $\mathbf{D}^0$ the matrix of distances between the loads in $\boldsymbol{\Omega}^0$ and $\boldsymbol{\Upsilon}^0$ (Figure 1). The time iterative process of the online clustering uses the core of K-means algorithm with:

- An assign step, where loads in $\boldsymbol{\Omega}^t$ are assigned to the closest centroid in $\boldsymbol{\Upsilon}^{t-1}$,
- Update centroids by averaging loads in each cluster.

It differs from K-means algorithm in using exponential smoothing to transfer structural information from previous time steps during the calculation of the

distance matrix,

$$\mathbf{\Delta}^t = \frac{\sum_{\tau=0}^{t} \lambda^{t-\tau} \mathbf{D}^\tau}{\sum_{\tau=0}^{t} \lambda^\tau} \tag{7}$$

λ is the exponential smoothing coefficient and takes a value in $[0, 1]$, which corresponds to how much of the previous time step information is transmitted to the next one. The assignment of the loads in $\mathbf{\Omega}$ is based on matrix $\mathbf{\Delta}$. Implementation of the exponential smoothing relies on the assumption that loads are relatively stable and those grouped in a same cluster may exhibit the same dynamic over time and thus remain together.

The centroids are updated by averaging the loads $\mathbf{X}_i^t \in \mathbf{P}_k^t$. Hence the most recent data are used in the calculation of the typical load profiles which provides up-to-date typical load profiles.

Consumption data display time-dependency. The clustering process presented in this work is distance-based, so the distance definition chosen should respect time dependency and create clusters based on pattern. Several distance measures using correlation coefficients, Euclidean distance and Dynamic Time Warping (DTW) have been used in clustering time series [16, 17]. To tackle this problem, we suggest to use a dissimilarity index [18],

$$d(\mathbf{X}, \mathbf{Y}) = \phi[\rho(\mathbf{X}, \mathbf{Y})] d_{DTW}(\mathbf{X}, \mathbf{Y}) \tag{8}$$

that balance a first order temporal correlation coefficient

$$\rho(\mathbf{X},\mathbf{Y}) = \frac{\sum_{m=1}^{M-1} (x_{m+1} - x_m)(y_{m+1} - y_m)}{\sqrt{\sum_{m=1}^{M-1} (x_{m+1} - x_m)^2} \sqrt{\sum_{m=1}^{M-1} (y_{m+1} - y_m)^2}}, \tag{9}$$

estimating the dynamic behaviors and the DTW distance

$$d_{DTW}(\mathbf{X},\mathbf{Y}) = \min_{r \in M} \left(\sum_{(i,j) \in \{1,\ldots,M\}^2} |x_i - y_j| \right), \tag{10}$$

for any given load $\mathbf{X}$ and typical load profile $\mathbf{Y}$ of same length M using the function ϕ

$$\phi(u)=\frac{2}{1+\exp(u)}, \tag{11}$$

an adaptive tuning function to balance automatically the distance d_{DTW} according to the temporal correlation coefficient ρ [19]. DTW has been broadly used in time series analysis and pattern recognition, it measures temporal similarities between time series. It calculates the Euclidean distance between a point of a time series and all the points of the other time series to create a distance matrix between each pair of points. It then finds the shortest way r from the lower left corner to the upper right corner which is called a warping path. We used the Paliwal window [20] to limit the shifting to only a few time steps (i.e. a window around the diagonal of the distance matrix) depending on the data resolution and to speed up the calculation.

3.3. Adaptivity: Semi-Online Facility Location

So far the algorithm is only online, but not yet adaptive. In this section, we present a probabilistic approach called facility location to adjust the number of clusters according to new unseen data [21]. When clustering loads, two antagonist processes have to be handled:

- Stability: loads changing shape simultaneously (e.g. seasonality) should stay in the same cluster,
- Novelty: disruptive load behavior should generate a new cluster center.

Facility location tackles the second point. A cost C_i^a is set to the assignment of each load, which is proportional to the distance d_{i,A_i} of each load $\mathbf{X}_i^t$ to

the closest typical load profile $\mathbf{Y}_{A_i}^{t-1}$ and a cost C^f to the creation of a new facility (i.e. cluster center) with $C^f \gg C_i^a$. C_i^a and C^f are combined to form a probability

$$p(\mathbf{Y}_{k+1}=\mathbf{X}_i)=\min\left(\frac{C_i^a}{C^f},1\right), \tag{12}$$

that a load becomes a new cluster center. Hence the larger d_{i,A_i}, the larger the marginal cost C_i^a of adding a new load to existing clusters, the higher the probability that it becomes its own cluster center. The facility cost is regulating the clusters' size and can be empirically chosen according to the clusters size wanted.

As the newly generated loads arrive simultaneously in a block $\mathbf{\Omega}$, facility location is implemented in a semi-online way and evaluates the creation of new cluster centers on $\mathbf{\Omega}$ instead of a load at a time as in the completely online setup [22]. A threshold γ_{min} is defined to set how many loads should exhibit a disruptive behavior to generate an extra typical load profile, thus sensitivity to outliers is reduced. Monte-Carlo simulations are run 1000 times to obtain the distribution of the number of loads above the threshold, and the mode of the distribution is used to evaluate if the threshold is reached or not. Hence it makes the algorithm consistent. If the threshold is reached, the load which is the farthest from its closest centroid is used as extra cluster center and the distance matrix $\mathbf{D}$ is then recalculated to check if other loads in $\mathbf{\Omega}$ should be assigned to the new typical load profile. Thereafter $\mathbf{\Delta}$ is also updated with $K+1$ clusters (Figure 1).

If two cluster centers are converging, the algorithm would ultimately merge them, but it would take many iterations. To avoid redundant typical load profiles, a minimum threshold $d_{min}<d(\mathbf{Y}_m,\mathbf{Y}_n)$ between two cluster

centers m and n is defined. If a pair of centroids gets under d_{min}, they are considered similar and are merged to form a single cluster with the average of their assigned loads as centroid. Thereafter $\mathbf{\Delta}$ is also updated with $K-1$ clusters (Figure 1). Hence the algorithm reacts faster to decreasing number of typical load profiles over time.

3.4. Setting the parameters

The online adaptive clustering algorithm requires setting parameters empirically as the nature of the data (e.g. electric load data, central heating district data), the resolution, the preprocessing affect the clustering process. Moreover the objective of the clustering can differ from one application to another, which is why the setting of the parameters is left to the user. Table 2 gives guidelines on the action of each parameter on the clustering process.

Table 2: Influence of the parameters on the clustering process.

Parameter	Definition	Value	Influence
K^0	Initial number of cluster centers	$n \in \{2,...,N\}$	On the first iterations
λ	Exponential forgetting	[0,1]	Smooth the clustering
C^f	Facility cost	relative to C^a	Size (radius) of the clusters
d_{min}	Minimum distance between cluster centers	Relative to the expected number of clusters	size (radius) of the clusters
γ_{min}	Number of disruptive load needed to create a new cluster	$n \in \{2,...,10\}$	Limits fluctuations of K

Setting K^0 has influence only on the few first iterations (depending on λ) as the initial structure will fade out progressively thanks to the exponential forgetting. λ is influencing how much structural information is transfered from one iteration to another, the larger λ the more information transmitted.

C^f and d_{min} are parameters controlling the adaptivity and more precisely controlling the radius of the clusters. A large value of C^f will allow larger clusters and thus a smaller K. A small value of d_{min} will allow the creation of smaller clusters and thus a larger K. γ_{min} can be used to limit fluctuation of K.

4. Online Adaptive Clustering Performance Evaluation Using Simulated Data

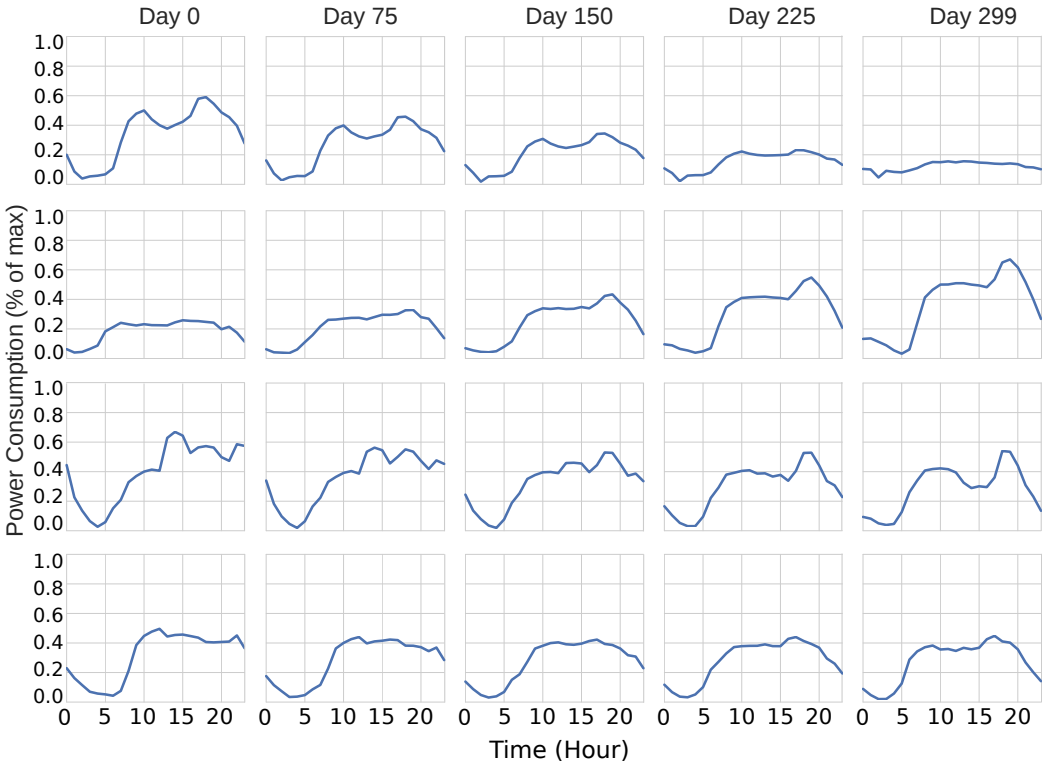
The performance of the online adaptive clustering algorithm is first evaluated on a synthetic dataset where the true typical load profile for each consumer is known. The following section describes how synthetic loads were generated, how the clustering process was evaluated and presents the results.

4.1. Data Generation

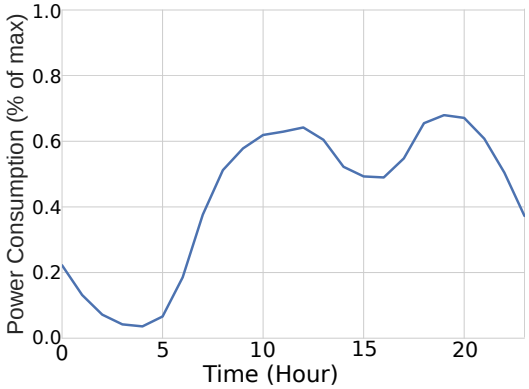
The following simulation aims at demonstrating that the proposed algorithm can both handle slow changing typical load profiles as well as disruptive behaviors (i.e. unobserved in the previous steps). To do so four slow changing and one disruptive typical load profiles are created based on nine typical load profiles from ENTSO-E data base displaying different load behaviors at hourly resolution [23]. The four slowly changing typical load profiles are generated using four pairs of typical load profiles from ENTSO-E. The daily typical load profiles $\mathbf{Y}_k$ are synthetically created as weighted averages

$$\mathbf{Y}_k^t = \left(1 - \frac{t}{300}\right)\mathbf{Y}_k^0 + \frac{t}{300}\mathbf{Y}_k^{299} \tag{13}$$

of $\mathbf{Y}_k^0$ and $\mathbf{Y}_k^{299}$ respectively the starting (Day 0) and ending (Day 299) profiles exhibited by typical load profile k. Figure 2(a) presents the evolution



(a)



(b)

Figure 2: Synthetic data generation based of ENTSO-E load profiles. Morphing of the four typical load profiles (a) and the new load behavior appearing at day 200 (b).

of these typical load profiles. From the four slow changing typical load profiles, 1000 individual daily loads are sampled. Each of the 1000 simulated customers is randomly assigned to one of the four slow changing typical load profiles $\boldsymbol{\Upsilon}=\{\mathbf{Y}_1,...,\mathbf{Y}_4\}$. The simulated daily loads

$$\mathbf{X}_i^t=\mathbf{Y}_k^t+\mathcal{N}(0,\boldsymbol{\Sigma}) \tag{14}$$

are sampled from the typical load profiles $\mathbf{Y}_k^t$ by adding a multivariate Gaussian noise $\mathcal{N}(0,\boldsymbol{\Sigma})$. The covariance matrix

$$\boldsymbol{\Sigma}=\begin{bmatrix} \sigma_1^2\rho_{11} & \dots & \sigma_1\sigma_l\rho_{1m} & \dots & \sigma_1\sigma_M\rho_{1M} \\ \vdots & \ddots & & & \vdots \\ \sigma_l\sigma_1\rho_{l1} & & \sigma_l^2\rho_{lm} & & \sigma_l\sigma_M\rho_{lM} \\ \vdots & & & \ddots & \vdots \\ \sigma_L\sigma_1\rho_{L1} & \dots & \sigma_L\sigma_m\rho_{Lm} & \dots & \sigma_L^2\rho_{LM} \end{bmatrix} \tag{15}$$

is stochastically generated. $\sigma=\{\sigma_1,...,\sigma_{24}\}$ is a normalized random vector of standard deviation and ρ is a matrix of coefficient decreasing exponentially from the diagonal

$$\rho_{lm}=\exp\left(\frac{-|l-m|}{\tau}\right), \tag{16}$$

which adds some small time shifting to the patterns. The daily load data were then normalized to [0,1] by dividing them by their reference power.

At time step 100, 250 of the 1000 customers generated disrupt from their slow transitioning typical load profile to the fifth typical load profile (Figure 2(b)). The number of clusters is in the same time expected to change automatically, generating an extra cluster center which groups the 250 disruptive loads. At time step 200, the 250 customers are catching back their

original slow transitioning typical load profiles. The number of clusters is then expected to change from five to four clusters and reassigning the loads to the cluster corresponding to their original typical load profile.

4.2. Performance Evaluation

The performance of the algorithm is evaluated using the Normalized Mutual Information (NMI) which is a criterion based on entropy and cross-entropy as defined in information theory. It actually evaluates how much information is shared between two vectors of labels. The entropy

$$\mathrm{H}(\mathbf{U})=\sum_{i=1}^{I} p(u_i)\log(p(u_i)), \tag{17}$$

is the amount of disorder in vector $\mathbf{U}$ where each element can take a value in $\{u_1,...,u_I\}$ and the probabilities $p(u_i)$ represent the probabilities that an object picked at random from $\mathbf{U}$ has value u_i. The Mutual Information (MI)

$$\mathrm{MI}(\mathbf{U},\mathbf{V})=\sum_{i=1}^{I}\sum_{j=1}^{J} p(u_i,v_j)\log\left(\frac{p(u_i,v_j)}{p(u_i)p(v_j)}\right) \tag{18}$$

is basically the entropy of the joint probability between vector of labels $\mathbf{U}=\{u_1,...,u_I\}$ and $\mathbf{V}=\{v_1,...,v_J\}$. In the context of this work, the joint probability $p(\mathbf{Y}_j,\mathbf{\Pi}_k)$ is the probability that a load is both from typical load profile $\mathbf{Y}_j$ and assigned to cluster $\mathbf{\Pi}_k$. MI is then normalized,

$$\mathrm{NMI}(\mathbf{U},\mathbf{V})=\frac{\mathrm{MI}(\mathbf{U},\mathbf{V})}{\sqrt{\mathrm{H}(\mathbf{U})\mathrm{H}(\mathbf{V})}} \tag{19}$$

and takes value in [0,1] where 1 is a perfect match between $\mathbf{U}$ and $\mathbf{V}$.

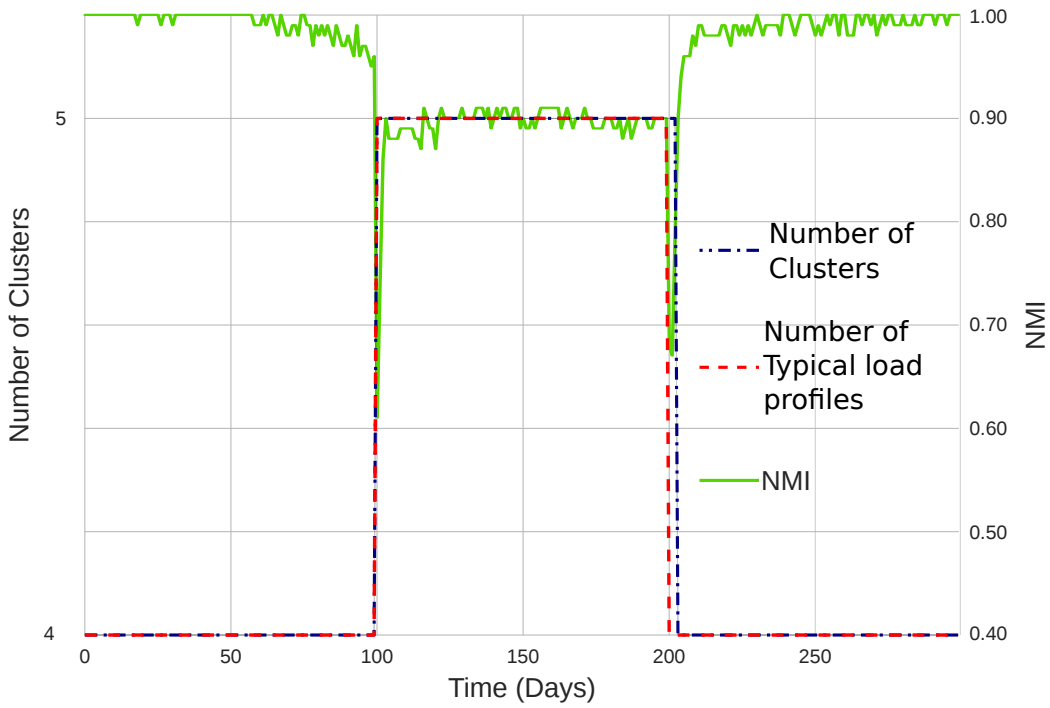


Figure 3: mode of the NMI and the number of clusters over the test period.

4.3. Online Clustering Setup and Results

The evaluation focuses on the online adaptive clustering algorithm. Hence the consensus clustering is not performed on the synthetic data and the first partition $\mathbf{\Pi}^0$ is actually the real assignment of the customers to typical load profiles in $\mathbf{\Upsilon}$. The facility cost is fixed to C^f=100, the number of points γ_{min} required to generate a new cluster is set to one, the exponential smoothing is λ=0.85 and the minimum distance between cluster centers is d_{min}=0.07.

The results of the analysis are presented in Figure 3 with the evolution of the number of clusters (blue dotted line) and the NMI (green solid line) over the test period. The number of typical load profiles used to sample the daily load curves is displayed with a red dashed line. It shows that the num-

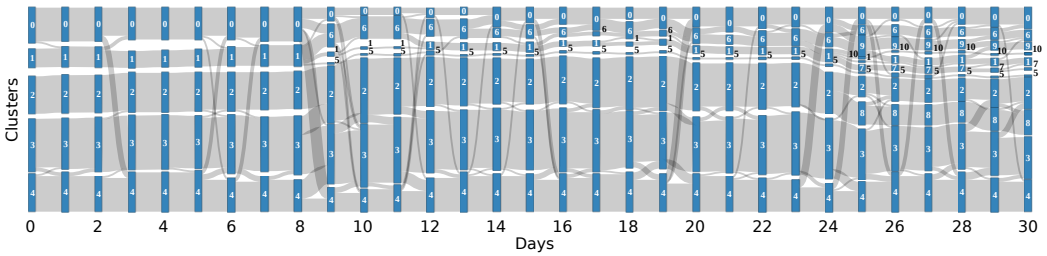


Figure 4: Sankey graph displaying the flow of buildings in the different clusters from one day to another during the test period.

ber of clusters generated by the online adaptive clustering algorithm follows precisely the number of typical load profiles until day 200. Even when at time step 100, the number of typical load profiles is changing from four to five. When the number of typical load profiles used to sample the loads is decreasing from five to four, we observe a delay of 2 days before the algorithm adjusts the number of clusters. Besides detecting correctly the number of clusters until day 100, the algorithm groups correctly the loads that are sampled from the same typical load profile in the same cluster as the NMI stays close to 1.0 until day 50. At day 50, a decrease of the NMI is observed. This can be explained by the real partition information provided at t=0 fading away in the exponential smoothing as well as a convergence of the typical load profiles (see Figure 2) which can engender some misclassifications. At day 100, we observe a drop to approximately 0.6 and the NMI comes quickly back to around 0.9 in the next day and oscillate around 0.9 until day 200. At day 200, the NMI drops again down to 0.75 and comes quickly back to 0.95 the next day and continues to increase until it oscillates between 0.99 and 1.0. At the end of the test period, the algorithm classifies correctly the loads despite the change from four to five and back to four typical load profiles.

The online adaptive clustering performed accurately on the synthetic dataset as it generates the correct number of clusters and groups correctly customers generated from the same typical load profile in the same cluster.

5. Applications to Real-world Data

The online adaptive clustering algorithm has been tested on two real-world datasets, i) central district heating loads from 97 buildings in Copenhagen at hourly resolution for a month, ii) 13241 electrical loads from industries, businesses and households with PV (i.e. for billing purposes) at hourly resolution for an entire year. They exhibit different characteristics which can be observed when profiling loads and demonstrate a wider range of applications of the online adaptive load profiling clustering to energy systems. The code of the online adaptive clustering algorithm has been made available on *GitHub*[1] to the interested reader.

5.1. Central District Heating Data

HOFOR, the operator of the central district heating in Copenhagen area, provided data from 97 building over a period of a month (31 days in Mach-April) with hourly resolution. A block $\mathbf{\Omega}$ groups a day of data (M=24) and no preprocessing has been operated on the data as the consumptions are all of the same magnitude [0,1].

The consensus clustering is using a modified version of the K-means algorithm that uses d_{DTW} as a distance metric and is applied with K_q={2,...,10}

[1]https://github.com/gleray/Online-Kmeans

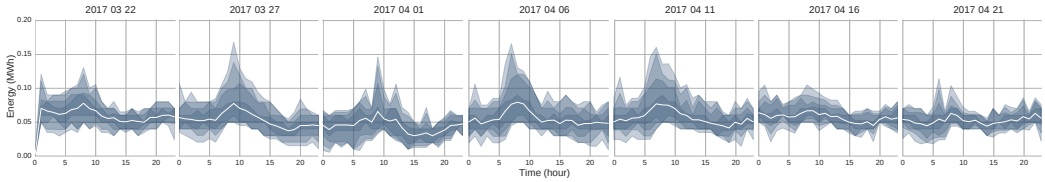


Figure 5: Evolution of cluster 4 along the test period.

on data generated at day 0. From the dendrogram obtained with the hierarchical ascending clustering, a partition $\mathbf{\Pi}^0$ into five clusters is generated. The online adaptive clustering uses the centroids $\mathbf{\Upsilon}^0$ and $\mathbf{D}^0$ obtained from $\mathbf{\Pi}^0$ as initial parameters and runs over 31 days.

The online adaptive clustering algorithm starts with K=5 clusters, a facility cost C^f=7.5, it needs only one building to create a new cluster center, the exponential smoothing is λ=0.85 and the minimum distance between cluster centers is d_{min}=0.01. The evolution of the clusters' composition is summarized in a Sankey diagram (Figure 4) that displays the flow of customers between clusters from one day to the next. The clusters are stable over time, besides some adjustments with few buildings changing cluster at each iteration. At iteration 9, cluster 1 splits into cluster 1, 6 and 5. Cluster 5 is actually a single building with a pattern different from the rest of the pool. We observe again a splitting at day 25, cluster 1 splits into cluster 1, 10, 7 and cluster 9 is taking elements from cluster 1, 0 and 6. The evolution of cluster 4 is presented in Figure 5 and shows how the typical load profile's shape changes slowly over the period.

No predefined classification is available to assess the quality of the partition, hence the number of clusters and the RMSE between individual loads and their assigned typical load profile over the test period (Figure 6) have

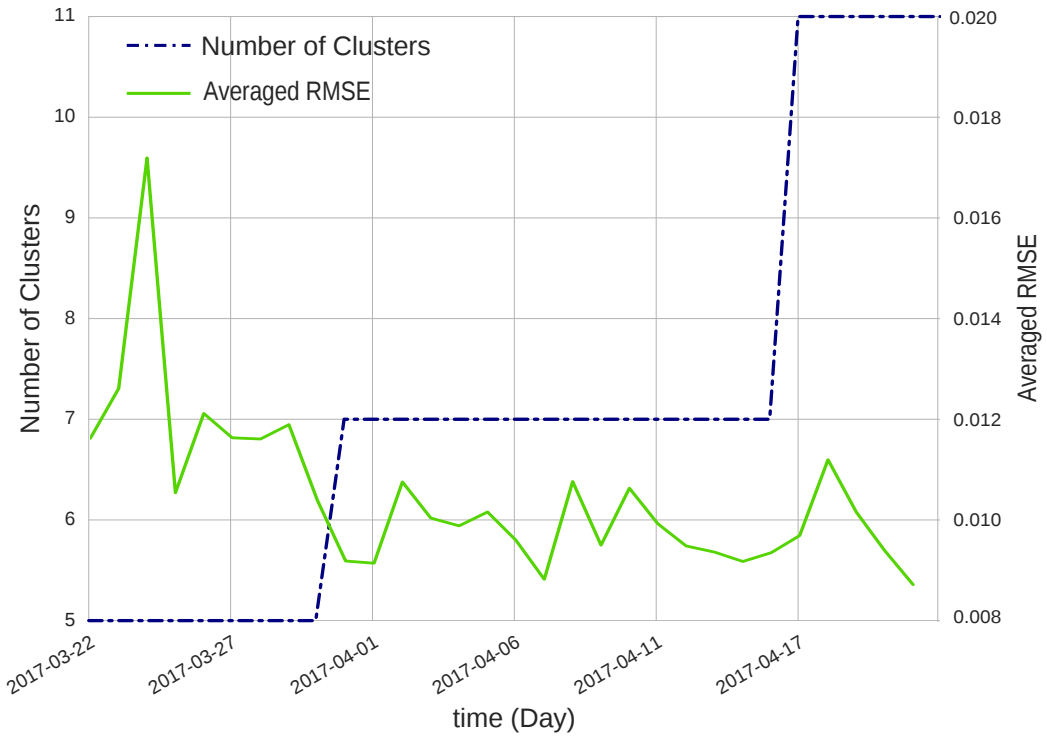


Figure 6: Number of clusters and Averaged Root Mean Square Error (RMSE) between daily loads and their assigned typical load profile over the test period for the central district heating data.

been used to evaluate the partition. As expected, the RMSE does not depend on the number of clusters and globally decreases over the period. It confirms first that the online adaptive clustering algorithm summarizes accurately the loads using a limited number of typical load profiles in the case of stable typical load profiles; second, it is fast as it uses only one K-means iteration at each time step and last but not least it demonstrates successful application of the online adaptive clustering algorithm for profiling non electric loads.

5.2. Electrical Load Data

Radius, the DSO in Copenhagen area, provided the second data set. It consists of a year (240 days after removing missing data) of hourly power consumption data from N=13241 customers. The customers metered at hourly resolution by Radius are businesses, industries and households with PVs. The blocks $\mathbf{\Omega}$ consist of a day (M=24) and the loads were preprocessed by dividing them by the peak over the period as explained in Section 2.

The consensus clustering is using the same modified version of the K-means algorithm presented in Section 5.1 with $K_q\{10,...,100\}$ on day 0: 2015-01-12. The partition $\mathbf{\Pi}^0$ is obtained by cutting the dendrogram into seven clusters. It is intentionally underestimating the number of clusters present in the dataset (approx. 20).

The online adaptive clustering algorithm starts with K=10 clusters centers, a facility cost set to C^f=950, the minimum number of customers to create a new cluster center is γ_{min}=5, an exponential smoothing coefficient of λ=0.85, and a minimum distance between two cluster centers set to d_{min}=0.13. A standard K-means algorithm using Euclidean distance and a Self Organizing Map (SOM) applied on the entire set of daily load pro-

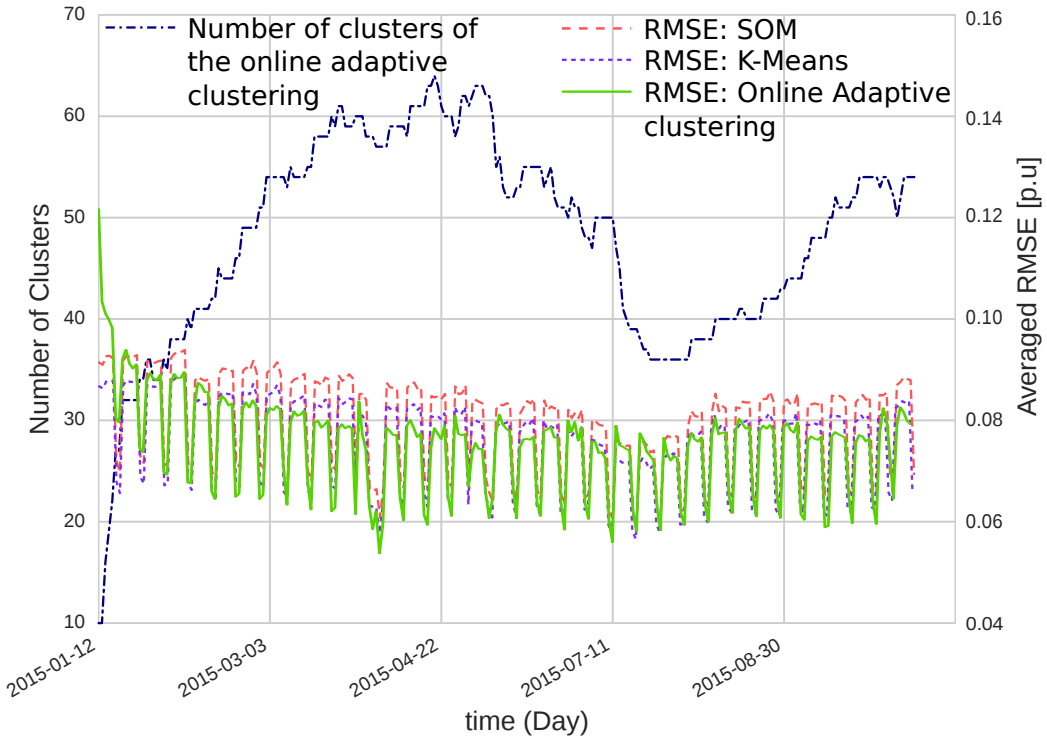


Figure 7: Number of clusters and Averaged Root Mean Square Error (RMSE) per unit of peak load, between daily loads and their assigned typical load profile over the test period for Benchmarks: SOM (7.9% RMSE overall score), K-means (7.5% RMSE overall score) and the online adaptive clustering algorithm (7.2% RMSE overall score).

files with K=50 have been used to benchmark the online adaptive clustering algorithm [5].

The results are presented in Figure 7 with the number of clusters and the averaged RMSE between daily loads and their assigned typical load profile for the benchmarks and the online adaptive clustering algorithm. The number of clusters generated by the online adaptive clustering algorithm first increases up to 64 around 2015-04-22 before going down to 36 during the summer and up again to 54. The averaged RMSEs show weekly periodicity which results

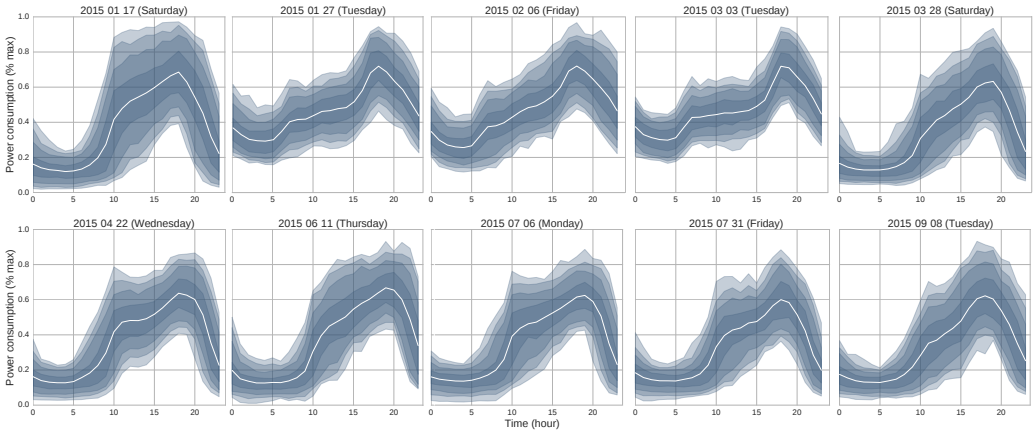


Figure 8: Evolution of cluster 15 along the test period.

from the customers' activity in the study, mostly industries and businesses, that are usually closed on weekends and thus displays more homogeneous patterns. The number of clusters seems to influence marginally the average RMSE of the online adaptive clustering algorithm, besides the first 14 days, which means that the algorithm is finding accurately the number of latent typical load profiles at each iteration. When the number of clusters in the online clustering algorithm gets close to 50, as set in the benchmark, the average RMSE of the benchmarks is getting closer (or equal) to the RMSE of the online adaptive clustering algorithm. The K-means actually beats the online adaptive clustering algorithm when the K^t is under 40. It stabilizes the RMSE between 0.08 (weedays) and 0.06 (weekends) when benchmarks have a tendency to fluctuate. After removing the 30 first days of convergence to a stable solution, the overall RMSE score of the online adaptive clustering algorithm is 7.2% against 7.5% and 7.9% respectively for the K-means and SOM. In terms of computational time, on a desktop computer equipped with Intel Xeon CPU 3.50 GHz $\times$ 8 cores, the consensus clustering (computed

in parallel on 7 cores) takes 15 minutes to be completed, and each iteration of the online adaptive clustering (on a single core) takes approximately 1 minute 20 seconds as the overall process on 240 days took 6 hours. As comparison, the SOM takes 16 hours to be completed and the K-means 20 hours. These results confirm the necessity of having an adaptive clustering process that estimates correctly the number of clusters in the data at time t. The difference of performance between the online adaptive clustering algorithm and the benchmark would be larger with a dataset displaying more non-stationary behavior like loads involved in Demand Response (DR) programs or with large amount of PVs.

Figure 8 gives an example of the evolution of a cluster over the period. The shape of the typical load profile as well as the lowest value are changing over the period. The cluster groups mostly restaurants, which are active daily from lunch to dinner time with a peak at dinner time and along the year at high activity periods (Christmas holidays, Saturdays, and from April to August).

From the application on real data the algorithm has fulfilled expectations, it handles correctly slow changing and fast changing profiles, splitting and merging to keep the same overall accuracy with a low computational cost at every iteration.

6. Conclusions and Future Works

In this paper, we have presented an online adaptive clustering methodology for load profiling. We have demonstrated its efficiency on both synthetic and real datasets. It is more agile than traditional clustering based load

profiling as it is recursive and uses the most recent data to update daily typical load profiles and computationally efficient as it processes only short time series at each time step of the online clustering. In this framework, the typical load profiles are generated every day, hence a customer's profile can be summarized by concatenating the typical load profiles assigned for each day to generate its load profile over a period of interest.

The methodology establishes a first step toward dynamic profiling of electricity consumption patterns. The deployment of smart meters associated with the increasing share of renewables will make dynamic load profiling compulsory for future grid management as fast decision making under uncertainty is becoming the common situation. A dynamic clustering methodology is then more suited for handling balancing between generation and consumption which is a dynamic problem both on production and demand sides.

From a widespread power system perspective, the methodology can be a systematic tool providing insights at a reasonable time scale for demand side management programs. It can also be used to develop new dynamic tariffs that would reflect the marginal cost each customer generates by shifting or synchronizing their peak load to the overall peak load. In electricity demand analysis a dynamic load profiling method will provide information on the stability of customers load patterns over time and whether it displays some periodicity (e.g. day open and close for a supermarket). Classic load profiling applications like estimating load for planning of the grid will also benefit from the output of the online adaptive clustering algorithm. Sampling of the most probable weekly or yearly load profiles can be generated by concatenating the daily load profiles and provides information on the uncertainty of the load

behavior *via* confidence intervals or probability density distribution around the curve.

Several possible extensions of the work can be considered; the methodology could be extended to multi-energy profiling which is lacking at the moment and will be needed as multi-energy solutions exploiting volatility of different markets are being deployed (e.g. fuel shifting solution). It could also be tested on a dataset with large amounts of non-stationary loads, typically households subject to DR or equipped with PVs, EVs and batteries. As the code has been made available on *GitHub*[2], we invite readers with such dataset to download the code and test it on their data. On a more technical aspect, the methodology could be transposed in a Bayesian framework, which will provide direct evaluation of the uncertainty.

Acknowledgment

The authors thank EnergyLab Nordhavn partners, Radius and HOFOR for supporting this work by providing data, as well as the EUDP for funding through the EnergyLab Nordhavn project (EUDP 64015-0055).

References

[1] EU. Directive 2009/72/EC of the European Parliament and of the council of 13 July 2009 concerning common rules for the internal market in electricity and repealing Directive 2003/54/EC, 2009.

[2] https://github.com/gleray/Online-Kmeans

[2] American Recovery and Reinvestment Act of 2009, Public Law 1115, 2009.

[3] Yi Wang, Qixin Chen, Chongqing Kang, Mingming Zhang, Ke Wang, and Yun Zhao. Load Profiling and Its Application to Demand Response: A Review. *Tsinghua Science and Technology*, 20(2):117–129, 2015.

[4] Kai-le Zhou, Shan-lin Yang, and Chao Shen. A review of electric load classification in smart grid environment. *Renewable and Sustainable Energy Reviews*, 24:103–110, 2013.

[5] Gianfranco Chicco, Roberto Napoli, and Federico Piglione. Comparisons among clustering techniques for electricity customer classification. *IEEE Transactions on Power Systems*, 21(2):933–940, 2006.

[6] Fintan McLoughlin, Aidan Duffy, and Michael Conlon. A clustering approach to domestic electricity load profile characterisation using smart metering data. *Applied Energy*, 141:190–199, 2015.

[7] I P Panapakidis, M C Alexiadis, and G K Papagiannis. Enhancing the clustering process in the category model load profiling. *IET Generation, Transmission Distribution*, 9(7):655–665, 2015.

[8] Kevin Mets, Frederick Depuydt, and Chris Develder. Two-Stage Load Pattern Clustering Using Fast Wavelet Transformation. *IEEE Transactions on Smart Grid*, 7(5):2250–2259, 2016.

[9] Sergio Valero Verdú, Mario Ortiz García, Carolina Senabre, Antonio Gabaldón Marín, and Francisco J García Franco. Classification, filtering,

and identification of electrical customer load patterns through the use of self-organizing maps. *IEEE Transactions on Power Systems*, 21(4): 1672–1682, 2006.

[10] Stephen Haben, Colin Singleton, and Peter Grindrod. Analysis and clustering of residential customers energy behavioral demand using smart meter data. *IEEE Transactions on Smart Grid*, 7(1):136–144, 2016.

[11] Ramon Granell, Colin J. Axon, and David C H Wallom. Clustering disaggregated load profiles using a Dirichlet process mixture model. *Energy Conversion and Management*, 92:507–516, 2015.

[12] Ignacio Benítez, Alfredo Quijano, José Luis Díez, and Ignacio Delgado. Dynamic clustering segmentation applied to load profiles of energy consumption from Spanish customers. *International Journal of Electrical Power and Energy Systems*, 55:437–448, 2014.

[13] Ignacio Benítez, José-Luis Díez, Alfredo Quijano, and Ignacio Delgado. Dynamic clustering of residential electricity consumption time series data based on Hausdorff distance. *Electric Power Systems Research*, 140:517–526, 2016.

[14] Charu C Aggarwal and Chandan K Reddy. *Data clustering: algorithms and applications*. CRC press, 2013.

[15] Thomas Grotkjær, Ole Winther, Birgitte Regenberg, Jens Nielsen, and Lars Kai Hansen. Robust multi-scale clustering of large DNA microarray datasets with the consensus algorithm. *Bioinformatics*, 22(1):58–67, 2006.

[16] T. Warren Liao. Clustering of time series data - A survey. *Pattern Recognition*, 38(11):1857–1874, 2005.

[17] John Paparrizos and Luis Gravano. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1855–1870. ACM, 2015.

[18] Ahlame Douzal Chouakria and Panduranga Naidu Nagabhushan. Adaptive dissimilarity index for measuring time series proximity. *Advances in Data Analysis and Classification*, 1(1):5–21, 2007.

[19] Alon Efrat, Quanfu Fan, and Suresh Venkatasubramanian. Curve matching, time warping, and light fields: New algorithms for computing similarity between curves. *Journal of Mathematical Imaging and Vision*, 27(3):203–216, 2007.

[20] Kuldip K Paliwal, Anant Agarwal, and Sarvajit S Sinha. A modification over sakoe and chiba's dynamic time warping algorithm for isolated word recognition. *Signal Processing*, 4(4):329–333, 1982.

[21] Adam Meyerson. Online facility location. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 426–431. IEEE, 2001.

[22] Edo Liberty, Ram Sriharsha, and Maxim Sviridenko. An Algorithm for Online K-Means Clustering. *CoRR*, abs/1412.5:1–13, 2014.

[23] ENTSO-E. Consumption Data Hourly load value of a specific country

for a specific day March 27th 2013, 2013. URL https://www.entsoe.eu/data/data-portal/consumption/Pages/default.aspx.