Two-stage load pattern clustering using fast wavelet transformation

Kevin Mets, Frederick Depuydt, Chris Develder, *Senior Member, IEEE*

***Abstract*—Smart grids collect large volumes of smart meter data in the form of time series or so-called load patterns. We outline the applications that benefit from analyzing this data (ranging from customer segmentation to operational system planning), and propose two-stage load pattern clustering. The *first stage* is performed per individual user and identifies the various typical daily power usage patterns (s)he exhibits. The *second stage* takes those typical user patterns as input, to group users that are similar. To improve scalability, we use fast wavelet transformation (FWT) of the time series data, which reduces the dimensionality of the feature space where the clustering algorithm operates in (i.e., from N data points in the time domain to $\log N$). Another qualitative benefit of FWT is that patterns that are identical in shape but just differ in a (typically small) time shift still end up in the same cluster. Furthermore, we use g-means instead of k-means as the clustering algorithm. Our comprehensive set of experiments analyzes the impact of using FWT vs. time-domain features, and g-means vs. k-means, to conclude that in terms of cluster quality metrics our system is comparable to state-of-the-art methods, while being more scalable (because of the dimensionality reduction).**

I. Introduction

The smart metering rollout as witnessed worldwide, enables detailed measurements of power consumption and production in real-time (or at least at a higher frequency than the still quite common yearly manual meter readings), and also enable, e.g., dynamic pricing schemes or other demand side management approaches. In this paper, we focus on the data analysis of such smart meter data obtained from distribution grid end-points (e.g., households, small businesses). The challenge of interpreting such a large volume of metering data is commonly tackled with machine learning techniques: in particular, clustering algorithms are used to group similar load patterns, which eases the analysis and interpretation of the metering data [1]. A “load pattern” represents the demand over a specific period of time, e.g., 24 hours with a resolution of 15 or 60 minute intervals. The profiling of such load patterns serves a broad range of applications, as discussed further in Section I-A.

We propose a hierarchical clustering scheme that comprises two major stages, as illustrated in Fig. 1: (i) The first stage analyses one user at a time and performs an unsupervised clustering of his/her behavior into groups of daily load patterns that are similar; (ii) The second stage takes the load patterns that represent each of the per-user clusters (i.e., the single centroid pattern for every per-user cluster), and thus identifies all possible daily load patterns seen across all users. (Note that in a practical implementation, the per-user clusterings in stage (i) can be executed in parallel, thus making this a potentially very fast scheme.[1]) Eventually, we end up with K global clusters, and each user can then be represented as a vector in a K-dimensional feature space, where element $n_{i,k}$ denotes the number of daily load patterns of user i that belong to global cluster k. In this paper, we investigate the features as well as the clustering algorithm to use for both load pattern clustering stages.

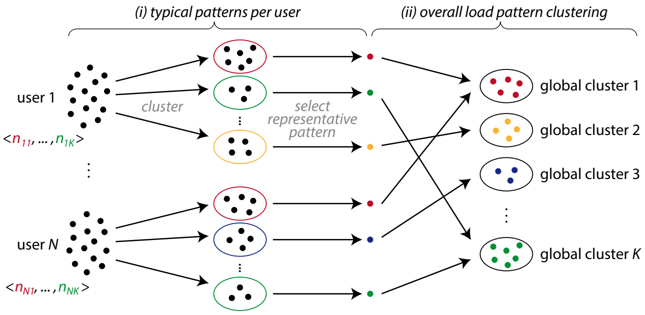

(i) typical patterns per user
(ii) overall load pattern clustering
user 1
$<n_{11}, \dots, n_{1K}>$
cluster
select representative pattern
user N
$<n_{N1}, \dots, n_{NK}>$
global cluster 1
global cluster 2
global cluster 3
global cluster K

Fig. 1. Our two-stage clustering approach.

In terms of features, we propose to use fast wavelet transformation (FWT) of the load pattern, instead of the raw time series itself (e.g., as used in [2]) or a fast Fourier transform (FFT) (as in [3]). For the clustering algorithm, we suggest to adopt the g-means algorithm [4], as an alternative for the common k-means algorithm. G-means automatically selects the optimal number of clusters, and makes less restrictive assumptions about the cluster shapes. In terms of dataset, we focus on a heterogeneous customer group connected to the low voltage distribution grid, as opposed to industrial medium voltage customers [1], [5] or nation-wide aggregated load data [6].

Our main contribution is the proposal of an *original combination* of (1) a hierarchical clustering approach (i.e., split into two stages), (2) adopting a wavelet transformation instead of working directly on the time domain data, and (3) g-means clustering instead of the more common k-means. While each of those components have been suggested by themselves, we

Manuscript received June 16, 2015. K. Mets was funded through a Ph.D. grant from the Agency for Innovation by Science and Technology (IWT). This work was supported by the Ministry of Science of Flanders via the project Linear organized by the IWT. K. Mets, F. Depuydt, and C. Develder are with with Dept. of Information Technology – IBCN, Ghent University – iMinds, Ghent, Belgium, e-mail: {firstname.lastname}@intec.ugent.be.

[1] In our non-optimized proof-of-concept Java implementation the entire two-stage clustering of 2 years of 15 min smart metering data from 244 users completes in only a few minutes, including the calculation of cluster quality metrics.

do believe we are the first to put these three pieces together. Moreover, we analyze — qualitatively and quantitatively (in terms of cluster quality metrics) — the impact of (a) using wavelet features vs. time domain features, as well as (b) using g-means instead of k-means clustering.

Now, we will first provide a comprehensive overview of the concrete applications that are served by profiling load patterns. Then, Section II presents an overview of the related work in load pattern clustering. In Section III we detail our proposed clustering approach, and subsequently analyze its performance through experiments in Section IV. Conclusions and future work are summarized in Section V.

A. Applications of load pattern clustering

Load profiling, load pattern clustering, segmentation, etc. are terms frequently used to describe the process of grouping similar load patterns into groups. Instead of describing related algorithms in detail (as done extensively in [1]), we comprehensibly present the applications that benefit from load profiling. Clustering or profiling of customer load patterns has many applications, including tariff design, load forecasting, power grid planning and operations, demand response, and energy efficiency programs.

1) Electricity demand analysis: From a high-level perspective, the power system consists of *generation*, *transmission*, and *distribution*. Load profiling is used to study demand patterns at these levels: e.g., nation-wide aggregated loads [6] (e.g., for generation planning), substations [7], distribution stations [8], large (industrial) customers [1], [9], [10], and more recently small residential or commercial customers [2].

2) Customer segmentation, tariffs, billing and markets: Currently, utility companies use demographic data (family size, house size, location, etc.) as the basis for customer segmentation and tariff design [11]. However, technological advances (e.g., smart meters), and liberalized energy markets lead to new possibilities in defining tariffs, exploiting detailed knowledge of customer's power consumption [1] to better suit the needs of both customers and utilities. The authors of [10], [12] propose load profiling for the purpose of pricing differentiation or designing demand response tariffs. Similar in spirit, [13] suggests load profiling information as the basis for billing of consumers that deviate from their contracted schedules. Load profiling can also be used to assist customers in selecting an adequate tariff [5].

3) Load forecasting: Load forecasting is an essential part of generation, grid operations, power markets, and regulation. Load estimation or forecasting forms the basis for system state estimation, which is used for power system planning (e.g., transformer, conductor sizing). Customer classification based on load patterns may provide relevant information for short-term and mid-term load forecasting. In [14], [15], the power system load is estimated by the aggregation of representative load patterns. For obtaining the latter, clustering is proposed [16]. Forecasting that relies on available data from similar days as the one the load is to be forecast for, clearly may benefit from load profiling to identify such similar days [17]. In [18] the authors use clustering for dis-aggregated electricity load forecasting. How exploiting customer grouping benefits the accuracy of load forecasting, in the context of market participation, is studied in [19]. The authors of [20] show that the power consumption of groups of customers with similar consumption patterns can be forecast more accurately than that of random customer groups.

4) Demand response and demand side management: Load profiling has been proposed as a means for enhancing targeting and tailoring of demand response and energy efficiency programs as well as improving energy reduction recommendations [2], [21]–[24]. Segmentation results are used to study the variability in energy consumption in [2]: e.g., it might be easier to target customers with low variability for demand response programs, and those with high variability for suggesting behavioral changes and energy efficiency programs. Load profiling can support the assessment of the impact of demand response programs [5], [16] and energy efficiency programs [25]. Regression models for demand reduction based on cluster analysis of load patterns are presented in [26]. In [27], load pattern data is used to infer occupancy states, to group users, and to determine demographic, household, and appliance stock characteristics. Clustering is used to identify controllable heating loads from smart meter data in [28].

5) Power system planning and operation: Load profiling may also be used to improve the planning and operation of the power system, especially in low-voltage networks, for which limited knowledge is available related to, e.g., the consumption patterns of households. Highly generalized (average) load patterns are typically used for decision making, but in reality domestic consumption patterns are highly diverse. Load flow simulations are often used as a tool for power system planning and operations, and in [11] the authors suggest the use of load profiling results to improve the accuracy of such power grid simulations: in a case study, maximum line currents and minimum node voltages were calculated both from realistically generated network load patterns (relying on clustering analysis) and those using the generalized patterns. The values obtained from simulations using the proposed clustering-based model proved to be very close to the real system values, as opposed to those using the generalized profiles. In [8], clustering is used for the analysis of the load patterns at distribution substations.

II. Related Work

We discuss three aspects of load pattern clustering: (i) first we present what features are typically used to represent the load pattern, i.e., the feature space in which the clustering algorithm is applied, (ii) subsequently the algorithm itself, and (iii) finally, the commonly defined metrics to evaluate cluster quality.

A. Features

The most commonly used features are the (daily) load pattern time series values themselves. These load patterns are typically preprocessed before clustering. A common step is to normalize every daily load pattern, e.g., (i) by rescaling it relatively between the minimum and maximum loads of the period under study [5], (ii) by dividing it by the user's reference power (e.g., peak power [1]), or (iii) by dividing it by its total consumption [2]. Such normalization methods are used to emphasize the shape of the load patterns, rather than to focus on the absolute value of their amplitude.

Preprocessing (e.g., normalization) is not only done on a per-pattern basis. The authors in [1] first group the daily load patterns according to predefined *loading conditions* (e.g., season, day) for the period under study. A single *typical daily load pattern* is then determined for each customer and loading condition: i.e., multiple daily load patterns are combined statistically (e.g., by taking the mean). Irregular patterns (e.g., peaks) thus will have a limited impact on the resulting typical daily load patterns (that will subsequently be clustered). In addition, the amount of data to be clustered is reduced significantly. However, because the thus defined "typical load patterns" are much smoother compared to the individual load patterns as measured, details and extremes of the behavioral patterns are lost and only general trends remain.

As alternative to the raw time domain data, some approaches use frequency domain features, based on a Fast Fourier Transform (FFT) [3], [29]. However, FFT is designed for stationary behavior, whereas load pattern time series over reasonably long time periods typically exhibit non-stationary behavior [6] (e.g., because of seasonal variations, or changes in the household composition in terms of family members and/or electric devices). In this setting, wavelet transformations are better suited; related work has also shown promising results when using wavelet-based features for clustering nation-wide aggregated power consumption patterns [6].

B. Clustering algorithms

A wide variety of approaches has been proposed for clustering load patterns: e.g., k-means, k-means++, fuzzy k-means, hierarchical clustering, modified follow the leader [1], Gaussian mixtures [11], self-organizing maps [3], [30]. We describe the k-means and g-means algorithms and refer to, e.g., [1] for a broader overview of the others.

1) K-means: Given a dataset $X = \{x_1, x_2, \ldots, x_N\}$ (with $x_i \in \mathbb{R}^n$), and an integer value K, the k-means algorithm partitions X in a set of K clusters $C = \{C_1, C_2, \ldots, C_K\}$. As output, each data point $x_i \in X$ is assigned to exactly one cluster $C_k \in C$, which is characterized by a cluster centroid μ_k. Starting from an initial set of cluster centroids $\{\mu_1, \mu_2, \ldots, \mu_K\}$, which are typically selected at random from X in classical k-means, the algorithm repeatedly executes two steps until convergence (i.e., the cluster members do not change anymore): (i) in the *cluster assignment step*, each data point x_i is assigned to the cluster with centroid μ_j that it is closest to; (ii) the *cluster update step* re-calculates the cluster centroid based on the current cluster members. For the assignment step (ii), typically "closest" is defined by the (squared) Euclidean distance:

$$\text{cluster}(x_i) = \underset{k \in \{1,\ldots,K\}}{\arg\min} \|x_i - \mu_k\|^2. \tag{1}$$

Thus, the clustering algorithm actually aims to minimize the *cost function* J (also called *distortion function*):

$$J = \frac{1}{N} \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2. \tag{2}$$

In the update step (ii), new cluster centroids are calculated as the average over current cluster members:

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i \tag{3}$$

The k-means algorithm is a simple and widely used algorithm. Nevertheless, it has certain drawbacks. The number of clusters K must be specified as input parameter, which is a non-trivial problem in many applications. The resulting clusters will be spherical because of the L2 cost metric, i.e., there is a prior assumption that the data is distributed in spherical clusters. The algorithm is also sensitive to the selection of the initial cluster centroids: the worst case running time is super-polynomial in the input size $|X|$, and the solution found may be a local optimum that is arbitrarily bad with respect to the optimization objective ($\min J$). To reduce the risk of ending up in a local optimum, the algorithm can be executed multiple times from which the best solution is selected (i.e., the one attaining the lowest J). Alternatively, k-means++ can be used to obtain a solution that better approximates the optimal J (i.e., with expected approximation ratio $\mathcal{O}(\log k)$) [31]: (i) select the first center μ_0 uniformly at random from X, (ii) select the next center randomly, where any remaining $x \in X$ is chosen with a probability proportional to $D(x)^2$, where $D(x)$ is the distance from x to the closest among the already selected cluster centers μ_j, (iii) repeat the previous step until K centers have been picked.

2) G-means: The g-means algorithm [4] can be seen as a wrapper around the k-means algorithm, that determines the optimal number of clusters and makes less assumptions about how data points are distributed within clusters (i.e., they do not necessarily need to be spherically distributed around the center). G-means iteratively increases the number of clusters as it progresses. In each iteration, every current cluster whose member data points do not appear to come from a Gaussian distribution is split into two. Between each round of splitting, k-means is executed on the entire dataset using the current cluster centroids to refine the solution. The statistical test, for checking whether data within a cluster is Gaussian, is based on the Anderson-Darling statistic, a powerful one-dimensional[2] normality test. Thus, instead of specifying K, the user only has to decide on the significance level α of the test. Yet, note that, if so desired, we could also run the same splitting procedure until reaching a given K (by iteratively splitting that cluster that adheres least to the Gaussian data distribution assumption).

[2] To reduce the data (n-dimensional vectors) to a single dimension, each point is projected on the line connecting the two candidate centers of the splitted clusters (e.g., along the first principal component of the data).

C. Evaluation criteria

Several evaluation criteria have been defined for the task of load pattern clustering, as detailed in [1] and summarized below. They rely on the following distance definitions:

- between two n-dimensional vectors (e.g., load patterns):

$$d(x_i, x_j) = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(x_{i,k} - x_{j,k})^2} \tag{4}$$

- between a vector x_j and a cluster C_j:

$$d(x_j, C_j) = \sqrt{\frac{1}{|C_j|}\sum_{x_k \in C_j} d(x_j, x_k)^2} \tag{5}$$

- within a set (e.g., cluster) C_j:

$$d(C_j) = \sqrt{\frac{1}{2|C_j|}\sum_{x_i \in C_j} d(x_i, C_j)^2} \tag{6}$$

The Mean Index Adequacy (MIA) is used to measure the compactness or homogeneity of the clusters:

$$\mathit{MIA} = \sqrt{\frac{1}{K}\sum_{k=1}^{K} d(\mu_k, C_k)^2}. \tag{7}$$

Other indicators consider not only the compactness of the cluster, but also the separation of clusters or distance between clusters. Noting the set of cluster centroids as R, the Cluster Dispersion Indicator (CDI) is:

$$\mathit{CDI} = \frac{1}{d(R)}\sqrt{\frac{1}{K}\sum_{k=1}^{K} d(C_k)^2}. \tag{8}$$

The Davies-Bouldin Index (DBI) represents the system-wide average of the similarity measures of each cluster with its most similar cluster:

$$\mathit{DBI} = \frac{1}{K}\sum_{k=1}^{K}\max_{i \neq j}\left\{\frac{d(x_i, C_k) + d(x_j, C_k)}{d(R)}\right\}. \tag{9}$$

The Similarity Matrix Indicator (SMI) is defined as the maximum off-diagonal element of the symmetrical similarity matrix, whose elements are computed using logarithmic function of the Euclidean distance between pairs of cluster centroids (i.e., representative load patterns):

$$\mathit{SMI} = \max_{\substack{i>j \\ i,j \in \{1,\ldots,K\}}}\left\{\left(1 - \frac{1}{\ln d(\mu_i, \mu_j)}\right)^{-1}\right\} \tag{10}$$

The ratio of "within cluster sum of squares to between cluster variation" (WCBR) is the ratio of the sums of the square distances between each input vector and its cluster's centroid vector and the distances between the clusters' centroids:

$$\mathit{WCBCR} = \frac{\sum_{k=1}^{K}\sum_{x_k \in C_k} d(\mu_k, x_k)^2}{\sum_{1 \leq q < p}^{K} d(\mu_p, \mu_q)^2} \tag{11}$$

Compactness and cluster separation show opposing trends, e.g., compactness increases with the number of clusters, but separation decreases. We will use these evaluation metrics as indicators of clustering quality in Section IV: the lower any of these metrics, the better.

III. Two-stage load pattern clustering algorithm

We have opted for a completely unsupervised approach, and thus do not rely on any a priori defined loading conditions, i.e., our input data only comprises the actual measured load patterns. Moreover, we jointly consider all types of low voltage customers, e.g., comprising residences as well as small businesses, for which it is difficult define a priori a single set of common loading conditions. For example, stores in Belgium are typically open on Saturday but not on Sunday, and have an additional closing day during the week: making an a priori distinction between weekdays and weekends does not necessarily make sense for all users. Further, we typically represent users with realistic, detailed behavioral patterns, i.e., we do not use profiles that are the mean of a set of actual patterns (which could be an acceptable representation of an aggregate of many such users). This is motivated by, e.g., demand response applications that may want to identify specific behavior (e.g., to target relevant customers, identify flexibility, steer demand), or energy efficiency applications that may target old or faulty appliances that exhibit irregular behavior. (We will provide samples of such mean profiles vs an actual single representative pattern in the experiments, Fig. 2 and 6). In terms of representing the load pattern, we have use wavelet-based features (as motivated in Section II-A), derived in a preprocessing stage detailed in Section III-A.

After that preprocessing, our clustering approach comprises two major stages illustrated in Fig. 1, following the same principle as [5]: (i) the *first stage* clusters the daily load patterns of a single user, and finds his/her representative patterns, while (ii) the *second stage* considers all user's representative patterns, which allows to group similar users. Both stages employ the same clustering algorithm to process the set of load patterns they take as input: the time series are converted to FWT-based feature vectors [6], which are then clustered using g-means [4]. Practical benefits of our approach are:

- The two-stage approach groups load patterns both on a per user level (leading to insights in personal behavior patterns), and on a user group level.
- Results from the first stage can be used for, e.g., load forecasting of customers, tariff selection, demand side management, energy awareness programs, etc. Results from the second stage can be used for power system planning, customer segmentation, tariff design, etc.
- We adopt a fully unsupervised strategy, and thus avoid a priori definition of loading conditions and corresponding typical load patterns, but instead derive everything from the data itself.
- We focus on grouping load patterns that exhibit similar behavior, and do not care so much about exact timing: we want to treat (slightly) time-shifted but otherwise similar shaped patterns as "similar".
- The selection of input data for stage two of the algorithm can be customized to the specific application.

More technical benefits are:

- The number of clusters is determined automatically using a single intuitive configuration parameter: the confidence level used to determine whether to split a cluster or not.[3]

- Features computed from the wavelet transform result in a significant reduction of the dimension of the feature vectors, thereby reducing computation time, memory, and storage requirements.
- Time invariance is reflected in the features, instead of handled by the clustering algorithm (e.g., the k-spectral centroids (KSC) algorithm used in [24]).
- The two-stage approaches improves scaling to large datasets: stage 1 can process all users in parallel[4], and limits the set of load patterns to cluster in stage 2.

A. Stage 0: Data pre-processing and feature extraction

Our aim is to cluster load patterns into compact and distinct groups. We deal with time series data representing energy consumption of low voltage customers, e.g., households or small businesses, usually obtained from smart meters. Such measurements (e.g., average net power consumption over a given interval) are typically performed on a 15 minute basis, resulting in load patterns of 96 samples per day (i.e., a 96-dimensional vector $\mathbf{x}$). We will transform those original feature vectors $\mathbf{x}$ to meet two objectives: (i) keep the feature space dimensionality compact, as to improve scalability (especially when smart metering would happen on even finer resolutions than our example of 15 min), and (ii) provide a degree of invariance to temporal translations (e.g., slight shifts in time), to focus on what behavioral patterns occur and less on their exact timing.

For the latter, it has been shown that using time domain features combined with the L2 norm (as adopted in many clustering algorithms) causes problems [24] (e.g., a "double penalty" is applied to patterns that are only slightly different in timing when peaks occur). While the clustering algorithm itself can adopt a more appropriate "similarity" notion that achieves a degree of time invariance (as is the basic idea of, e.g., the k-spectral centroids algorithm (KSC) used in [24]), we achieve a similar result purely by our choice of features.

Instead of using the time series data itself (as is the common approach), we convert it to a lower dimensional representation (i.e., with fewer features). Inspired by [6] we adopt the Fast Wavelet Transform (FWT) with Haar wavelets to transform a time series data vector $\mathbf{x}$. Wavelets capture the general trend of the input data in an approximation component, while the localized changes are kept in the detail components. Wavelet representation describes the time series in both time and frequency domain. However, before the FWT can be applied, the time series must be upsampled to obtain $N = 2^L$ samples (e.g., using linear interpolation): in our example, we transform a 96-dimensional original time series vector $\mathbf{x}$ to a 128-dimensional upsampled vector $\mathbf{x}^{\mathrm{U}}$. Next, we apply a FWT transformation W_ψ, to obtain $\mathbf{x}^{\mathrm{W}}$, describing the variations at different temporal scales (frequencies) $(d_0, \ldots, d_{L-1})$ and a constant approximation term c_0. At each scale d_j, the data is observed with a finer resolution (at higher frequency), i.e., in 2^j points. Thus, we have [6]:

$$W_\psi : \mathbb{R}^N \to \mathbb{R}^N, \mathbf{x}^{\mathrm{U}} \to \mathbf{x}^{\mathrm{W}} \triangleq (c_0, d_0, \ldots, d_{L-1}) \qquad (12)$$

$$\text{where } \mathbf{d}_j \triangleq (d_{j,0}, \ldots, d_{j,2^j-1}) \qquad (13)$$

From the FWT result $\mathbf{x}^{\mathrm{W}}$, we extract the coefficients for each scale and calculate the energy therein to obtain a feature vector $\mathbf{f} = (f_0, \ldots, f_{L-1})$ (i.e., the "absolute contributions" representation of [6]):

$$f_j \triangleq ||\mathbf{d}_j||^2 \quad j \in \{0, 1, \ldots, L-1\} \qquad (14)$$

Thus, detailed timing information is removed, and the feature vector dimensions are reduced (in our example from $N = 96$ for $\mathbf{x}$, to only $L = 7$ for $\mathbf{f}$). An important consequence of disregarding the approximation term c_0 is that the features will be invariant to vertical shifts of the load patterns (i.e., adding/subtracting a constant value to/from the whole load pattern time series) — still, if the application so requires, the approximation term of the wavelet coefficients can also be included, leading to a feature vector $\mathbf{f}' = (c_0, f_0, \ldots, f_{L-1})$. Finally, we perform range normalization, i.e., from $\mathbf{f}$ to a normalized $\mathbf{f}^{\mathrm{N}}$ with each $f_j^{\mathrm{N}} \in [0, 1]$, before using them as input to the clustering algorithm.

B. Stage 1: Typical load patterns per user

The first stage comprises two steps: (i) clustering of the daily patterns of individual customers, and (ii) selection of representative patterns (to be used as input for Stage 2). In a certain sense this step corresponds to a data driven unsupervised alternative to the a priori definition of loading conditions and derivation of typical load patterns.

1) Individual user clustering: This stage treats each customer individually, i.e., takes as input the dataset of daily load patterns from a single consumer (which is assumed to have been preprocessed to extract the features and prepare for clustering as laid out in Section III-A). Clustering is performed using the g-means algorithm. Note that individual customers can easily be clustered independently and in parallel. Loading conditions can be determined from these results, and used as selection criteria for stage 2 inputs.

2) Selecting representative patterns: We will use selected representative patterns as input for stage two. In [5] the cluster centroid from the largest cluster is chosen in the context of setting tariffs. Instead, we have chosen to select a representative pattern from each of the user's clusters to keep as much information as possible, and thus represent a user by a set of patterns rather than just a single one. We thus have a more detailed and complete view of a user compared to [5]. Further, we still adopt the wavelet features for such a representative pattern, as opposed to the time domain representation used by [5].

C. Stage 2: Overall load pattern clustering

The input for stage 2, after range normalization, is clustered using the same approach as in stage 1 of the algorithm. The resulting clusters group similar patterns from multiple users, after which each user can be represented in terms of how many of his day patterns belong to what global cluster.

[3] As pointed out before, if one really wants to specify the number of clusters exactly, we can still accommodate that.

[4] Note that also the FTW features can be calculated independently for each daily pattern and thus in parallel.

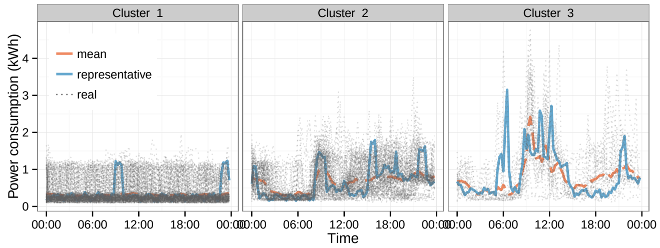


Fig. 2. The clusters as found in Stage 1 for a single user, with $\alpha = 0.01\%$.

IV. Experimental results

We evaluate the two-stage load profiling algorithm on a dataset covering one year of 15 min resolution power consumption measurements at Belgian user connection points in the low voltage distribution network. After filtering out customers that did not have a full year of data (e.g., equipment failures, or because the smart meters were installed later in the year), we retained a set of 244 users (that include both residences as well as small businesses, e.g., shops).

In Section IV-A we analyze stage one of the algorithm in detail, in terms of (i) number of clusters per customer, (ii) wavelet vs time domain features, and (iii) k-means vs g-means. In Section IV-B we report cluster quality metrics for the complete two-stage algorithm output. We recall that cluster quality metrics are those defined in Section II-C.

A. Stage 1: Typical load patterns on a per user basis

1) Example user: Figure 2 shows an illustrative result of the clusters obtained after stage one of the algorithm using a significance level $\alpha = 0.01\%$ in the g-means algorithm: this user exhibits three distinct patterns. Choosing larger values for α will result in more clusters and thus a more detailed view, however low values are useful to determine the general trends as we do here. The first cluster contains load patterns without much activity (e.g., vacation): low consumption with few peaks. Although the peaks occur at different times, the pattern is similar. This demonstrates the time invariance built into our wavelet features. The second cluster groups the general demand patterns of the household. The third group comprises patterns that exhibit high demand around midday, which we found to occur primarily on Wednesdays in spring and summer.[5] Finally, we point out that the "average pattern" (i.e., the cluster centroid as often used by others) obviously has a fairly smooth curvature and thus would not be representative of any single real pattern (including our "representative pattern"), which is more spiky.

2) Number of clusters per user: One of the benefits of g-means is that it automatically determines the number of clusters, which will depend on the setting of the significance level α (lower α means fewer clusters). Figure 3 and Table I show the histogram and average of the number clusters per user for varying α. These results also compare the number of clusters among both wavelet based and time domain features.

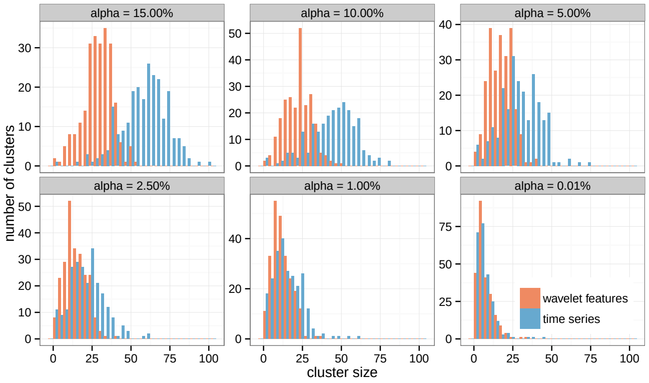


Fig. 3. Histograms of cluster sizes per user (obtained after *stage 1*) for different significance level parameter values α used by the g-means algorithm: the y-value is the number of users (out of the 244) that exhibit a number of clusters amounting to the x-value. (Note that the Y-axes have different ranges for different α.)

Table I
Average number of clusters per user obtained after *stage 1*.

	significance level α					
	15%	10%	5%	2.5%	1%	0.01%
Wavelet	28	21	15	12	10	6
Time	54	41	27	20	14	6

As intuitively expected, using time series features typically results in more clusters per user, given that wavelet representations incur some time shift insensitivity (as was our aim) and inherently have lower dimensionality. But for decreasing α, the distributions of the number of clusters per user converges for both feature representations and almost coincide for $\alpha = 0.01$.

3) Comparison between time and wavelet based clustering: We now analyze the difference between using time series features vs wavelet based features. We have just shown that the number of clusters obtained with g-means and a given α may lead to significantly different number of clusters for wavelet vs time domain features (Fig. 3). Thus, we will compare clustering results with the same number of clusters k to focus on the influence of the data representation. We adopt the approach discussed in [29] for comparing results obtained from different feature sets.

We proceed as follows, for each feature set individually: (i) we first use g-means algorithm using different values for the significance level α to determine the number of clusters for each user, and then (ii) use these cluster counts as input for the k-means algorithm. We then calculate the cluster quality metrics over each stratum of user profile clusters with k clusters (where a particular cluster may be the result of different α for different users), where profiles are represented in the time domain (to allow direct comparison of the metric for time- vs wavelet based clusters). Figure 4 shows the average performance for user profile clusters with the same k, for both feature types. The time series based features lead to the best result (i.e., lower values for the evaluation criteria). This should be no surprise because the evaluation criteria

[5] This possibly indicates a family with children home on Wednesday, yet we do not have metadata asssociated with the households, so this is mere speculation.

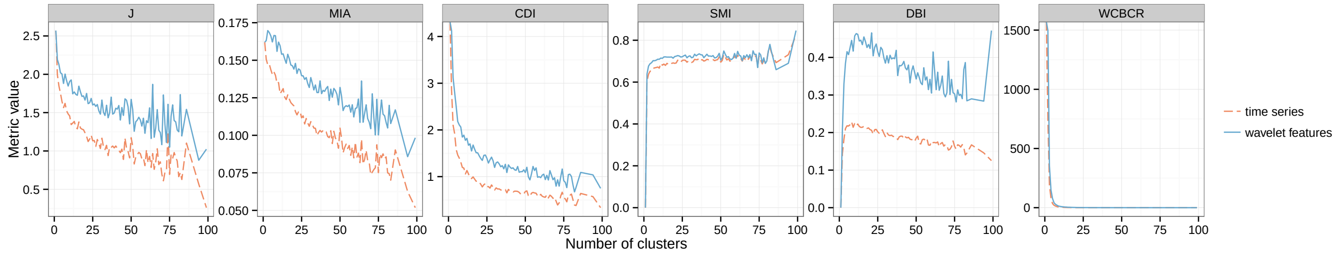
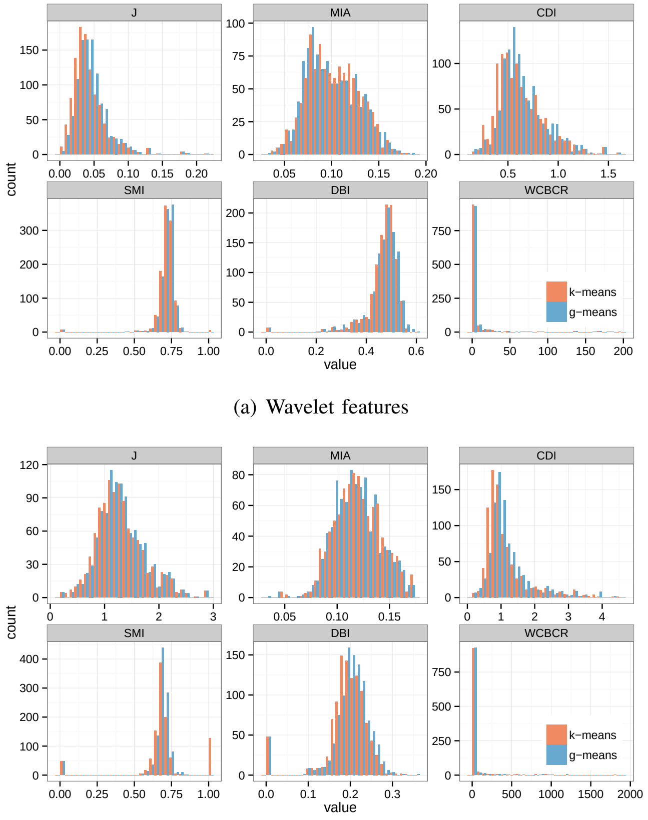


Fig. 4. Comparison of the time and wavelet based approaches. Performance metrics are compared in the time domain. Lower values correspond to higher quality clusters.



(a) Wavelet features

(b) Time domain features

Fig. 5. Histograms comparing the cluster quality indicators for the g-means and k-means algorithm. The y-value of a bar represents the number of cases where the quality indicator amounted to the x-value.

are designed for and calculated using time series data: the distance measures in that domain penalize grouping time-shifted patterns (as in wavelet based clustering). Nevertheless, the quality metrics for wavelet based clusters demonstrate very similar trends.

4) K-means vs. g-means: To compare the performance of the g-means algorithm with the k-means algorithm[6], we proceed as follows: (i) we first cluster the load patterns of a single user with g-means, and (ii) then take the number of clusters found by the g-means algorithm as input for k-means. We calculate various cluster quality indicators for both algorithms, over the clusters found per user. Figure 5(a) shows the histogram of the cluster quality indicator values over the whole user population for both g-means and k-means algorithms, when using wavelet features. Figure 5(b) shows the same for clustering with time domain features. We observe that the results are very similar for both g-means and k-means, independent of the features used. Still, the prior assumptions made by the respective algorithms are different, thus results are not exactly the same. The k-means algorithm assumes that data points in each cluster are spherically distributed. More generally, the g-means algorithm assumes a multi-dimensional Gaussian distribution (while in practice testing that assumption along the first principal component).

B. Stage 2: Typical load patterns of a user group

To analyze the complete two-stage algorithm, Fig. 6 shows the ten largest clusters obtained from stage two, visualized in the time domain. Performance metrics are listed in Table II for different values of α. Cluster 8 could be considered a commercial cluster (e.g., shop or office) because most consumption occurs during typical office hours. The remaining clusters correspond to patterns associated to different types of residential customers.

1) Number of clusters: We first look at the final number of clusters obtained after stage two, using wavelet based features for clustering. Table II shows the number of clusters obtained after stage one and two for different hyper-parameter choices, using the same α for both stages of the algorithm. We used the cluster centroids from the individual users as input data for stage 2. Note that one may opt to use different significance levels for stage 1 and stage 2: e.g., a high significance level may be specified for stage 1 to maintain a more detailed view of individual behavior. However, it is difficult to specify an optimal choice for the significance level and therefore the significance level should be chosen in function of the target application of the clustering results.

2) Clustering performance evaluation: Table II provides values for the evaluation criteria defined in Section II-C for different choices of the confidence level α. We do not provide a comparison to a time domain feature set because the two-stage approach and g-means clustering algorithm would make comparisons difficult (e.g., different number of clusters).

[6] The results presented are the best solution over 100 distinct k-means runs with random initializations. We validated that using k-means++ led to the same results.

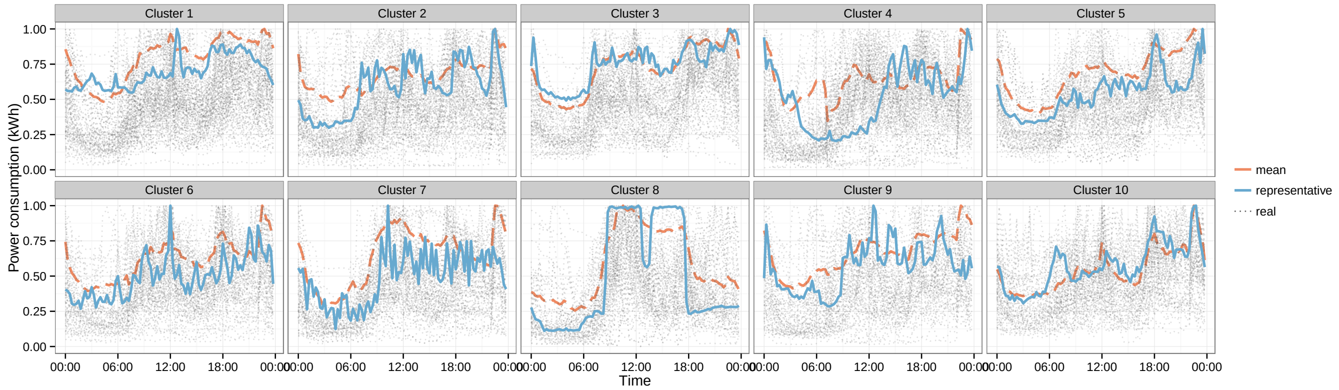


Fig. 6. Ten largest stage two ($\alpha = 1\%$) clusters visualized in time domain, normalized to emphasize load pattern shape.

Table II
Performance indicators of the two-stage algorithm using wavelet based features.

	# Clusters		Cluster Quality Metric for Stage 2					
α	Stage 1	Stage 2	J	MIA	CDI	SMI	DBI	WCBCR
15.00%	7045	476	0.017	0.050	0.269	0.815	0.502	0.002
10.00%	5486	291	0.020	0.054	0.299	0.801	0.505	0.006
5.00%	4261	175	0.025	0.057	0.327	0.795	0.519	0.016
2.50%	3523	112	0.028	0.058	0.350	0.795	0.515	0.041
1.00%	2892	112	0.027	0.055	0.338	0.800	0.490	0.034
0.01%	1775	68	0.033	0.058	0.334	0.821	0.495	0.060

However, the results presented in Section IV-A3 provide more information on the impact of feature extraction and selection.

Note that the absolute values of various performance metrics for the global representative pattern clusters are lower (thus better clusters, in terms of homogeneity and/or separation) than for the per-user clusters (recall Fig. 5(a)). A possible reason is that the representative patterns (i.e., cluster centroids from stage 1) that we cluster in stage 2 are more smooth: they exhibit lower variance as opposed to the single user original patterns.

V. Conclusions and Future Work

We have introduced a two-stage approach for clustering daily load patterns of a group of low voltage distribution grid customers: (i) first cluster similar daily load patterns per user to find each user's typical daily patterns, (ii) then cluster the complete set of typical user patterns over all users together. We propose to perform the clustering in a space of lower dimensionality (than the original time domain) by transforming the time series data using fast wavelet transformation. In addition, we advocate for the use of the g-means clustering algorithm rather than the popular k-means (e.g., with optimized initialization, such as k-means++).

We apply it on a set of load patterns from individual users — as opposed to earlier work using a similar (single phase) strategy, that considered aggregated consumption patterns [6]. (Note that our case forms a more challenging task because of the larger variety and variability in load patterns.) This analysis of power consumption of individual customers can be used to provide feedback on, e.g., energy consumption, tariffs selection, and load forecasting (e.g., for demand response). The main benefits of our approach can be summarized as:

- The clustering process is entirely unsupervised and thus does not need the a priori definition of a representative day or mean day as suggested by other authors [5]. Moreover, the strategy for selecting the patterns to cluster in stage 2 (e.g., mean profiles per stage 1 cluster versus an individual representative one) can be adapted to the specific application.
- The two-stage approach leads to a more scalable system. Stage 1 can be performed in parallel for each user, and stage 2 operates only on representative data from stage one, rather than the complete consumption data of all customers: alternative methods instead execute a clustering algorithm on the full dataset.
- That scalability of the clustering step is also improved by the feature dimension reduction (from N to $\lg N$, e.g., from 96 time measurements per day to a 7-dimensional vector) through fast wavelet transformation. An additional benefit thereof is that the proposed transformation results in grouping of similar patterns, even if they are slightly time-shifted.
- By using g-means (instead of, e.g., k-means++), the number of clusters is automatically picked based on setting an intuitive[7] parameter, the significance level (α) for the statistical test used by the g-means algorithm. (Yet, the clustering algorithm can still be applied to obtain a given number of K clusters as well.)

The contribution of our work also pertains to studying the

[7] We note that in practice, it may take some experimental tries to determine the most suitable α. Post hoc inspection of the results, i.e., of how "similar" the patterns within individual clusters are, and/or of the number of clusters obtained may be necessary to qualitatively validate the clusters are meaningful for the application at hand.

impact of the wavelet representation (vs. the original time domain data) and the g-means algorithm (vs. k-means):

- The results of using wavelet based features show the same trend compared to clustering in the time domain, despite the strong dimensionality reduction (e.g., with a factor of more than 10×, from 96 to 7). We have confirmed this both qualitatively and quantitatively. (For the latter, we stress that in absolute values, metrics are calculated in the time domain, which penalizes the insensitivity to temporal shifts as exhibited by our wavelet representation: that results in lower absolute values of the performance metrics [24].)
- The cluster quality metrics from the the g-means algorithm's clusters show similar distributions and values as those of k-means clusters. Thus, g-means proves to be a viable alternative, with a more flexible and intuitive configuration (cf. the significance level α).

Future work comprises incorporating the proposed two-stage approach into demand response (DR) applications, e.g.:

- Improved load forecasting of individual customers and/or a group of customers (taking the cluster analysis as input).
- Detect and predict *flexibility* in power consumption, to analyze the potential of DR and eventually exploit it. (A starting point would be taking the insights in typical consumption patterns from the clustering analysis, e.g., to compare "business-as-usual" behavior with the effect of adopting a DR program.)
- Automatically learn flexibility, from minimally intrusive measurement data, reducing or even eliminating the need for manual user input (e.g., avoid users to have to explicitly indicate flexibility through manual configuration).

References

[1] G. Chicco, "Overview and performance assessment of the clustering methods for electrical load pattern grouping," *Energy*, vol. 42, no. 1, pp. 68–80, Jun. 2012.

[2] J. Kwac, J. Flora, and R. Rajagopal, "Household energy consumption segmentation using hourly data," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 420–430, Jan. 2014.

[3] S. Verdu, M. Garcia, C. Senabre, A. Marin, and F. Franco, "Classification, filtering, and identification of electrical customer load patterns through the use of self-organizing maps," *IEEE Trans. Power Syst.*, vol. 21, no. 4, pp. 1672–82, Nov. 2006.

[4] G. Hamerly and C. Elkan, "Learning the k in k-means," in *Proc. Adv. Neural Information Processing Syst. (NIPS)*, Vancouver and Whistler, BC, Canada, 8–13 Dec. 2003, pp. 1–8.

[5] G. Tsekouras, N. Hatziargyriou, and E. Dialynas, "Two-stage pattern recognition of load curves for classification of electricity customers," *IEEE Trans. Power Syst.*, vol. 22, no. 3, pp. 1120–28, Aug. 2007.

[6] A. Antoniadis, X. Brossat, J. Cugliari, and J.-M. Poggi, "Clustering functional data using wavelets," *Int. J. Wavelets Multiresolution Inf. Process.*, vol. 11, no. 1, pp. 1–30, Jan. 2013.

[7] M. Espinoza, C. Joye, R. Belmans, and B. De Moor, "Short-term load forecasting, profile identification, and customer segmentation: a methodology based on periodic time series," *IEEE Trans. Power Syst.*, vol. 20, no. 3, pp. 1622–30, Aug 2005.

[8] E. Bobric, G. Cartina, and G. Grigoras, "Clustering techniques in load profile analysis for distribution stations," *Adv. Electr. Comput. Eng.*, vol. 9, no. 1, pp. 63–66, 2009.

[9] L. Hernández, C. Baladrón, J. M. Aguiar, B. Carro, and A. Sánchez-Esguevillas, "Classification and clustering of electricity demand patterns in industrial parks," *Energies*, vol. 5, no. 12, pp. 5215–28, 2012.

[10] T. Zhang, G. Zhang, J. Lu, X. Feng, and W. Yang, "A new index and classification approach for load pattern analysis of large electricity customers," *IEEE Trans. Power Systems*, vol. 27, no. 1, pp. 153–160, Feb. 2012.

[11] B. Stephen, A. Mutanen, S. Galloway, G. Burt, and P. Jarventausta, "Enhanced load profiling for residential network customers," *IEEE Trans. Power Delivery*, vol. 29, no. 1, pp. 88–96, Feb. 2014.

[12] H. Liu, Z. Yao, T. Eklund, and B. Back, "Electricity consumption time series profiling: A data mining application in energy industry," in *Proc. 12th Ind. Conf. Adv. Data Mining: Applications and Theoretical Aspects (ICDM)*, Berlin, Germany, 13–20 Jul. 2012, pp. 52–66.

[13] D. Gerbec, S. Gasperic, I. Smon, and F. Gubina, "Allocation of the load profiles to consumers using probabilistic neural networks," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 548–555, May 2005.

[14] C. Chen, M. Kang, J. Hwang, and C. Huang, "Synthesis of power system load profiles by class load study," *Int. J. Power Energy Syst.*, vol. 22, no. 5, pp. 325–330, 2000.

[15] J. Jardini, C. M. V. Tahan, M. Gouvea, S. U. Ahn, and F. M. Figueiredo, "Daily load profiles for residential, commercial and industrial low voltage consumers," *IEEE Trans. Power Delivery*, vol. 15, no. 1, pp. 375–380, Jan. 2000.

[16] G. J. Tsekouras, F. D. Kanellos, V. T. Kontargyri, I. S. Karanasiou, A. D. Salis, and N. E. Mastorakis, "A new classification pattern recognition methodology for power system typical load profiles," *WSEAS Trans. Cir. and Sys.*, vol. 7, no. 12, pp. 1090–04, Dec. 2008.

[17] Y. Chen, P. Luh, C. Guan, Y. Zhao, L. Michel, M. Coolbeth, P. Friedland, and S. Rourke, "Short-term load forecasting: Similar day-based wavelet neural networks," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 322–330, Feb. 2010.

[18] M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi, "Optimized clusters for disaggregated electricity load forecasting," *REVSTAT Stat. J.*, vol. 8, no. 2, pp. 105–124, Nov. 2010.

[19] P. Goncalves Da Silva, D. Ilic, and S. Karnouskos, "The impact of smart grid prosumer grouping on forecasting accuracy and its benefits for local electricity market trading," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 402–410, Jan. 2014.

[20] S. Patel, R. Sevlian, B. Zhang, and R. Rajagopal, "Aggregation for load servicing," in *Proc. IEEE PES General Meeting*, National Harbor, MD, USA, 27–31 Jul. 2014.

[21] I. Dent, T. Craig, U. Aickelin, and T. Rodden, "An approach for assessing clustering of households by electricity usage," in *Proc. 12th Ann. Workshop Comput. Intell. (UKCI)*, Edinburgh, Scotland, 5–7 Sep. 2012.

[22] B. A. Smith, J. Wong, and R. Rajagopal, "A simple way to use interval data to segment residential customers for energy efficiency and demand response program targeting," ACEEE Summer Study on Energy Efficiency in Buildings, Tech. Rep., 2012.

[23] H.-A. Cao, C. Beckel, and T. Staake, "Are domestic load profiles stable over time? An attempt to identify target households for demand side management campaigns," in *Proc. IEEE Ind. Electron. Soc. (IECON)*, Vienna, Austria, 10–13 Nov. 2013, pp. 4733–38.

[24] A. Albert, T. Gebru, J. Ku, J. Kwac, J. Leskovec, and R. Rajagopal, "Drivers of variability in energy consumption," in *Proc. ECML-PKDD Data Anal. Renew. Energy Integr. (DARE)*, Prague, Czech Republic, 23 Sep. 2013.

[25] A. M. S. Ferreira, C. A. M. T. Cavalcante, C. H. de Oliveira Fontes, and J. E. S. Marambio, "Pattern recognition of load profiles in managing electricity distribution," *Int. J. Ind. Eng. Manage. (IJIEM)*, vol. 4, no. 3, pp. 117–122, 2013.

[26] N. Yamaguchi, J. Han, G. Ghatikar, S. Kiliccote, M. Piette, and H. Asano, "Regression models for demand reduction based on cluster analysis of load profiles," in *Proc. IEEE PES/IAS Conf. Sustainable Altern. Energy (SAE)*, Valencia, Spain, 28–30 Sep. 2009, pp. 1–7.

[27] A. Albert and R. Rajagopal, "Smart meter driven segmentation: what your consumption says about you," *IEEE. Trans. Power Syst.*, vol. 28, no. 4, pp. 4019–30, Nov. 2013.

[28] H. Niska, "Extracting controllable heating loads from aggregated smart meter data using clustering and predictive modelling," in *Proc. 8th IEEE Int. Conf. Intell. Sens., Sens. Netw., Inf. Process. (ISSNIP)*, Melbourne, Australia, 2–5 Apr. 2013, pp. 368–373.

[29] E. Carpaneto, G. Chicco, R. Napoli, and M. Scutariu, "Customer classification by means of harmonic representation of distinguishing features," in *Proc. IEEE Power Tech. Conf.*, Bologna, Italy, 23–26 Jun. 2003, pp. 1–7.

[30] G. Chicco, R. Napoli, F. Piglione, P. Postolache, M. Scutariu, and C. Toader, "Load pattern-based classification of electricity customers," *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 1232–39, May 2004.

[31] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA 2007)*, New Orleans, Louisiana, 7–9 Jan. 2007, pp. 1027–1035.

Kevin Mets received the M.Sc. degree in Computer Science from Ghent University, Ghent, Belgium, in 2009. Until 2015 he was a member of the research group IBCN of the Dept. of Information Technology (INTEC) at Ghent University – iMinds, Ghent, Belgium, where he obtained a Ph.D. degree in Computer Science. His research focused on smart grids, optimization, demand side management algorithms for electric vehicles, and smart grid analytics. In 2015 he joined Sentiance as a data scientist, where he works on behavioral profiling and real-time context detection using mobile sensor data.

Frederick Depuydt received his M.Sc. degree in electronics from KaHo Sint-Lieven, Ghent, Belgium, in 2001. He worked in hardware prototyping and micro-electronics before receiving his M.Sc. in Computer Science from Ghent University, Ghent, Belgium, in 2008. He is currently working as a researcher at the research group IBCN of the Dept. of Information Technology (INTEC) at Ghent University – iMinds, Ghent, Belgium.

Chris Develder is professor with the research group IBCN of the Dept. of Information Technology (INTEC) at Ghent University – iMinds, Ghent, Belgium. He received a M.Sc. degree in computer science engineering and a Ph.D. in electrical engineering from Ghent University, in July 1999 and Dec. 2003 respectively (as a fellow of the Research Foundation, FWO). From Jan. 2004 to Aug. 2005, he worked for OPNET Technologies, on optical network design and planning. In Sep. 2005, he rejoined INTEC as a postdoctoral researcher, and as a postdoctoral fellow of the FWO since Oct. 2006 (until 2012). In Oct. 2007 he obtained a part-time, and since Feb. 2010 a fulltime professorship at Ghent University. He has stayed as a research visitor at UC Davis (Jul.-Oct. 2007), CA, USA and at Columbia University, NY, USA (2013-14).

Chris currently leads two research teams within IBCN, one on information retrieval and extraction, the other on smart grids. His research interests also still include optical networks (dimensioning, modeling, optimization, esp. for grid/cloud computing). He regularly serves as reviewer/TPC member for international journals and conferences (IEEE/OSA JLT, IEEE/OSA JOCN, IEEE/ACM Trans. Networking, Computer Networks, IEEE Network, IEEE JSAC, IEEE Commun. Mag., IEEE Trans. Smart Grids; IEEE Globecom, IEEE ICC, IEEE SmartGridComm, ECOC, ACM SIGIR, ACM CIKM, etc.). He is Senior Member of IEEE and Member of ACM.