Unsupervised Energy Disaggregation: From Sparse Signal Approximation to Community Detection

Pierre Winkler, Guillaume Le Ray, *Student Member, IEEE,* and Pierre Pinson, *Senior Member, IEEE*

***Abstract*—Data collected from smart meters is to be potentially used as a basis for many different purposes e.g. demand response management and pricing. A popular problem to tackle using such smart meter data is Non-Intrusive Load Monitoring (NILM), which is a deconvolution process to separate appliances and their consumption based on aggregated measurements only. State-of-the-art NILM approaches are supervised algorithms based on e.g. Factorial Hidden Markov Models (FHMM) or Artificial Neural Networks (ANN). They require large training datasets and may have limited generalization ability when used for new data and situations not seen during the training process. In contrast here, we propose an unsupervised NILM approach that combines (i) sparse signal approximation into a sum of boxcar functions by Orthogonal Matching Pursuit (OMP), (ii) a Gaussian Mixture Model (GMM) reducing redundancies in the boxcar functions, (iii) community detection to obtain appliance signatures from association of boxcar functions. The algorithm shows performance in the same range as the FHMM and NN at high resolution (6 seconds) but can also perform well at lower resolution (1 minute). As the approach is generic and unsupervised, it fits the requirements for a real-world implementation with standard metering data.**

***Index Terms*—Clustering, Community Detection, Non-Intrusive Load Monitoring (NILM), Orthogonal Matching Pursuit (OMP), Unsupervised Learning.**

I. Introduction

LARGE scale deployment of smart meters and, more generally, information and communication technologies (ICT) open up towards intelligent networks that will comprise the informational backbone of future smart grids. A prospect is to benefit from Demand Side Management (DSM) concepts to integrate Renewable Energy Sources (RES) efficiently and economically by harnessing consumer flexibility. With that objective in mind, smart meters connecting consumers and utilities (plus possibly some other agents in the system e.g. flexibility aggregators) with a two-way communication protocol allow utilities to get near real-time feedback on electricity consumption, while consumers can receive incentives e.g. dynamic tariffs, and modify their energy consumption behavior accordingly.

The exponentially growing amount of metering data brings new perspectives in terms of solving challenging R&D problems while allowing for new business models. These generally relate to developing new insights about electric demand characteristics relevant to DSOs and aggregators (e.g. consumer behavior, detection of electric vehicles and heat pumps, estimation of PV capacity), to the consumers (more control on electricity consumption) and to electricity providers (energy efficiency programs) [1]. Non-Intrusive Load Monitoring (NILM), which consists in separating single appliances and their electricity consumption signal from the overall consumption signal of a household thanks to their specific signatures, is probably the most known approach to extract information on what happens behind the meter. NILM was first described by Hart in the 90's [2] but applications became only possible with the booming of high frequency metering and the increase of computing resources as the process potentially requires heavy computational resources.

Hidden Markov Models (HMM) [3] and Artificial Neural Networks (ANN) [4] are the two dominant approaches used in the NILM literature and thus form the state-of-the-art [5]. However they are both supervised approaches which present certain drawbacks: (i) they are computationally expensive to train, (ii) a representative training (labeled) dataset should be available, and (iii) they cannot cope with new appliances (unseen during the training). Hence they perform well on overall electricity consumption signals that consist of the sum of appliance consumption signals observed during the training phase and that have a data granularity high enough to detect transient signatures. These are fairly strong restrictions for real-world implementation [5].

A few examples of unsupervised NILM approaches using event detection can be found in the literature [6], [7]. However they can only disaggregate relatively simple appliances with on/off states or multi-states with strong dependency on the labeling process. Another unsupervised approach consists in generating an over-complete dictionary of boxcar functions using a sparse decomposition algorithm to approximate the overall consumption [8]. Thereafter the boxcar functions are then labeled using a classifier which is trained on single appliances consumption patterns.

In contrast with the existing methods, we propose here an unsupervised NILM approach that takes advantage of (i) a sparse decomposition algorithm based on Orthogonal Matching Pursuit (OMP) principles, as presented in [8], (ii) a Gaussian Mixture Model (GMM) to reduce statistical redundancy of boxcar functions in the dictionary, and (iii) a community detection approach to obtain appliance signatures from association of boxcar functions. The performance is first benchmarked against existing methods (FHMM, ANN) on the UK-DALE dataset [9]. The aim of the benchmarking is to compare the performance of our algorithm against state-of-

P. Winker, G. Le Ray and P. Pinson are with the Centre for Electric Power and Energy, Technical University of Denmark, Kgs. Lyngby, Denmark (email: {gleray,ppin}@elektro.dtu.dk).

This work was partly supported by EUDP through the EnergyLab Nordhavn project (EUDP 64015-0055).

the-art methodologies. However, as it is not fair to compare supervised and unsupervised approaches, on such grounds only, we do not expect to outperform then, but at least to reach similar performance in an unsupervised environment. We subsequently evaluate how the performance of our approach is affecting by the granularity i the data collected. This is done by changing the resolution from six seconds to one minute. Having access to data at the minute resolution, or coarser, is more in line with real-world implementation that can be envisaged today.

The remainder of this paper is organized as following. After this introduction, Section II gives the overall sketch of the methodology as well as its individual components. Then, Section III concentrates on its application on the UK-DALE dataset, with both a benchmarking exercise (against state-of-the-art supervised approaches) at 6 seconds, and an analysis of the performance obtained when degrading the temporal resolution in the data at hand. Eventually, we gather a set of conclusions and perspectives for future work in Section V.

II. Methodology

The energy disaggregation problem can be mathematically formulated as

$$x^t = \sum_{n=1}^{N} y_n^t + \varepsilon^t, \quad \forall t \tag{1}$$

where y_n^t is the individual consumption of appliance n considered in the model and ε^t are the residuals which correspond to the error and the consumption of appliances not considered in the model [10]. Our approach to unsupervised NILM is illustrated in Figure 1, where the various blocks involve coarse decomposition approximation, dimension reduction, community detection and labelling. These are all covered individually and successively in the following.

A. Power Signal Sparse Approximation

Power signal sparse approximation uses a large set functions φ stored into a dictionary $\mathbf{\Phi} = \{\varphi_1, \ldots, \varphi_k, \ldots, \varphi_K\}$ to approximate the aggregated power consumption (top rectangle in Figure 1). The observation of load behavior and the over-representation of type I (i.e. ON/OFF) and type II (i.e. multistate) appliances dictate the choice of translation-invariant boxcar functions to fill the dictionary [8]. For each index k, the boxcar function is characterized by two parameters l and w, such that

$$\varphi_k^t = \frac{1}{\sqrt{w}} \, \Pi_{l-w/2, l+w/2}^t, \tag{2}$$

where l is the translation of the boxcar function, w the width and $\Pi_{a,b}^t = H(t-a) - H(t-b)$, where H is a Heaviside step function. The general shape of a boxcar function and the different parameters involved in the design are presented in Figure 2.

In a power signal sparse approximation framework, (1) is reformulated as

$$x^t = \sum_{k=1}^{K} \alpha_k \varphi_k^t + \varepsilon^t, \quad \forall t \tag{3}$$

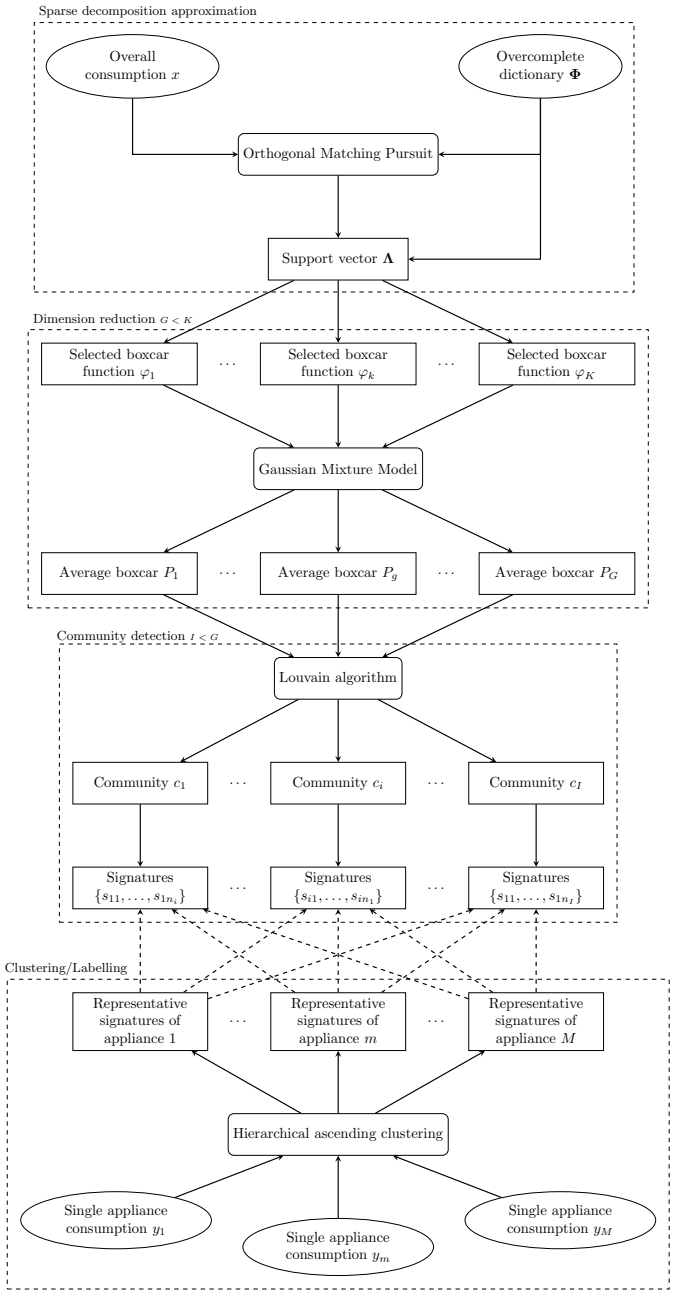

Fig. 1. Algorithm overview.

where α_k denotes the activation coefficient of the boxcar function φ_k. A compact vectorial form of the equation is, $\mathbf{x} = \boldsymbol{\alpha}\,\mathbf{\Phi} + \boldsymbol{\epsilon}$ where $\mathbf{x}$ is the load vector of all time indices, $\boldsymbol{\alpha}$ is the vector of coefficients α_k and $\boldsymbol{\epsilon}$ a vector of residuals.

The overcomplete dictionary $\mathbf{\Phi}$, with all the possible boxcar functions, is generated ahead of the approximation [11]–[13]. Hence only a subset of J boxcar functions with $J \ll K$ is used during the approximation. Subsequently, $\boldsymbol{\alpha}$ is a sparse matrix as most of the coefficients are equal to zero. A support vector Λ regroups the non-zero entries in $\boldsymbol{\alpha}$ and restricts the dictionary $\mathbf{\Phi}$ to the subset $\mathbf{\Phi}_\Lambda$. The aim of our implementation is to adjust sparsity by setting a fixed or an adaptive threshold corresponding to the minimum variation of power to consider as a change of state, in order to eliminate transient states or

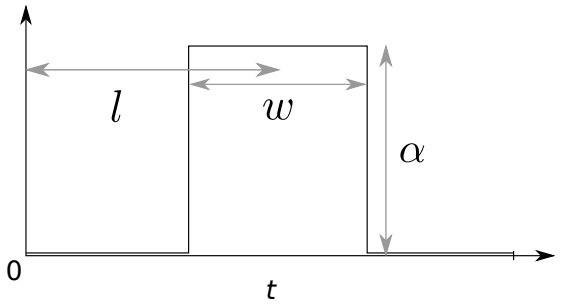

Fig. 2. Representation of the general shape of a boxcar function

internal states fluctuation of appliances.

Direct approaches to sparse approximation, like combinatorial optimization, are complex and require important computing resources to perform the approximation. A greedy algorithm using an iterative process is implemented instead [14]. Matching pursuit, the most used greedy algorithm, finds locally the optimal solution that is also close enough to a global optimal solution at each iteration. Orthogonal Matching Pursuit (OMP) outperforms matching pursuit by updating all activated coefficients simultaneously in generating the orthogonal projection of the selected boxcar functions at every iteration [15]. However OMP requires more computational resources due to extra calculations.

At iteration $j = 0$, the residuals are set to $\mathbf{r}_0 = \mathbf{x}$, the coefficients are set to $\boldsymbol{\alpha}_0 = 0$ and the support vector Λ_0 is empty. At each iteration j, a single element k_j that maximizes

$$k_j = \underset{k}{\operatorname{argmax}} ||\mathbf{r}_{j-1}\varphi_k||_2, \tag{4}$$

and corresponds to the element of the dictionary with the largest width w that fits under $\mathbf{r}_{j-1}$. The last selected function k_j is then added to the support vector

$$\Lambda_j = \Lambda_{j-1} \cup k_j. \tag{5}$$

The coefficients $\boldsymbol{\alpha}_j$ are computed as least square estimates, i.e,

$$\boldsymbol{\alpha}_j = \underset{\alpha}{\operatorname{argmin}} ||\mathbf{x} - \boldsymbol{\alpha}_{\Lambda_j}\boldsymbol{\Phi}_{\Lambda_j}||_2 = \boldsymbol{\Phi}^{\dagger}_{\Lambda_j}\mathbf{x}, \tag{6}$$

where $\boldsymbol{\Phi}^{\dagger}_{\Lambda_j}$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{\Phi}_{\Lambda_j}$ [16]. The approximated signal $\widehat{\mathbf{x}}_j$ is then updated, $\widehat{\mathbf{x}}_j = \widehat{\mathbf{x}}_{j-1} + \alpha_{k_j}\varphi_{k_j}$ where α_{k_j} is the activation coefficient of φ_{k_j}. Finally, the residuals are updated as $\mathbf{r}_j = \mathbf{x} - \widehat{\mathbf{x}}_j$. The output is the support vector Λ_J restricting the overcomplete dictionary to $\boldsymbol{\Phi}_{\Lambda_J}$.

B. Dimension Reduction

After sparse signal approximation, the load signature of an appliance is formed by a combination of boxcar functions which are alike. Hence these boxcar functions appear close to each others at every activation event. They have a strong cross-time dependence. But the OMP selects almost all boxcar functions only once and the community detection relies on the assumption of cross-time dependency of boxcar functions to form the multi-state signatures. To create redundancy and cross-time dependencies between boxcar functions generated by the same appliance, a clustering is implemented on the J selected boxcar functions. The input of the clustering is the parameters $(\boldsymbol{\alpha}, \mathbf{w})$ that describe the shape (power amplitude, operation time) of boxcar functions. The shape of clusters in the 2D space $(\boldsymbol{\alpha}, \mathbf{w})$ is not known *a priori*. A Gaussian mixture model fits 2D Gaussian distributions to form the clusters which means that they can have round or ellipsoidal shape in the 2D space depending on whether their covariance matrix structure [17]. Selected boxcar functions in Λ_J have coordinates $\zeta_j = (\alpha_j, w_j)$ and all the points form $\boldsymbol{\zeta} = \{\zeta_1, \ldots, \zeta_J\}$. A mixture of G clusters is then defined as

$$p(\boldsymbol{\zeta}) = \sum_{g=1}^{G} \phi_g \, \mathcal{N}(\boldsymbol{\zeta} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g), \tag{7}$$

where $\boldsymbol{\mu_g}$ is the mean, $\boldsymbol{\Sigma_g}$ is the covariance matrix and ϕ_g is the mixture coefficient of the g^{th} cluster. The probability density function $p(\boldsymbol{\zeta})$ integrates to one with $\phi_g \geq 0$ and $\mathcal{N}(\boldsymbol{\zeta}|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \geq 0$. The mixture coefficients ϕ_g are constrained by

$$\sum_{g=1}^{G} \phi_g = 1 \quad \text{and} \quad 0 \leq \phi_g \leq 1. \tag{8}$$

Boxcar functions in the same cluster are considered as the same boxcar function in the community detection. G, the number of clusters, is determined empirically with backwards and forward fine-tuning as the performance of the community detection depends on G.

C. Community Detection

The GMM generates cross-temporal dependencies between boxcar functions that are parts of the same signature. Graph theory is a field of mathematics that analyzes relationships (connections) between objects in a network (graph). A graph $\Gamma(V, E)$ is a mathematical representation of the pairwise relations between objects, where interactions are expressed with a set of vertices (i.e. boxcar functions) V and a set of edges (cross-temporal dependency) $E = \{e(u, v) \ : \ u, v \in V\}$ [18]. The relations $e(u, v)$ and $e(v, u)$ can be considered different (edges are oriented in a direction) or equivalent (edges are undirected or bidirectional). In this work we consider that $e(u, v) = e(v, u)$.

Community detection consists in forming strongly interconnected subsets from the graph [19]. In the context of this work, a community is a set of boxcar functions with strong cross-temporal dependencies as they appear repeatedly adjacent.

A weight $\omega_{u,v}$ sampled from a Gaussian distribution is assigned to each edge $e(u, v)$

$$\omega_{u,v} = \exp\left(\frac{-(t_u - t_v)^2}{2\sigma^2}\right), \tag{9}$$

σ denotes the scaling parameter and t_u and t_v are respectively the position of the center of boxcar functions u and v. It measures the strength of the cross-temporal dependency between u and v.

Concretely, the community detection algorithm forms disjoints groups $\{c_1, \ldots, c_i, \ldots, c_I\}$ that regroup boxcar functions with strong cross-temporal dependencies. The core of

the community detection process relies on the definition of the objective function that evaluates the aggregation into communities. The objective function defines the notion of community as groups of vertices with better internal connections than external [20]. The most used objective function for community detection is the modularity Q of the partition

$$Q = \sum_{i=1}^{I} \left[\frac{\Sigma_{in}^{c_i}}{2m} - \frac{{\Sigma_{tot}^{c_i}}^2}{4m^2} \right], \tag{10}$$

where m is the sum of all the weights in the network, $\Sigma_{in}^{c_i}$ is the sum of the weights from the internal edges of community c_i and $\Sigma_{tot}^{c_i}$ the sum of the weights from the edges incident to a vertex in community c_i [21]. Q takes values in $[-1, 1]$ where a high value reveals strong interconnections between the vertices in the same community and less dense connections between the vertices of the neighboring communities.

Computing the modularity of the communities is an NP-complete optimization problem and thus computationally expensive. The Louvain method is a heuristic approach: in practice it is iterative and a greedy-type algorithm (inspired by hierarchical ascending clustering) which can solve the problem in $O(n \log n)$ [19], [20]. As in the hierarchical ascending clustering, the first phase starts with each vertex u as its own community and then for each vertex the algorithm calculates the gain of modularity of moving u from its community to the community c_i of a neighbor vertex v

$$\Delta Q_{u,c_i} = \frac{\sum_{v \in c_i} \omega_{u,v}}{2m} - \frac{\Sigma_{tot}^{c_i} \omega_u}{2m^2}, \tag{11}$$

where, ω_u is the sum of the weights of the edges incident to the vertex u, $\sum_{v \in c_i} \omega_{u,v}$ is the sum of the weights of the edges from the vertex u to vertices in community c_i (only one v at this stage). u gets assigned to the community that maximized $\Delta Q_{u,c_i}$ only if it is positive. This stage is operated until the communities get stable, when no individual move can improve the modularity [19]. The second step builds a network of the communities obtained after convergence, using the sum of the weights of the edges between vertices of the communities. Then the first step is reapplied on the network and so on until no improvement on the modularity is observed.

D. Labelling Using Clustering

The disaggregation is unsupervised which means that the appliance consumptions have been individualized but not identified. Hence a post-processing labelling of the signatures in each community is necessary to identify them. Appliances have generally several programs (i.e. cycles of a fridge or programs of a washing machine) which means that an appliance exhibits many signatures. To identify the different signatures after disaggregation, we propose to generate sets of signatures from the single appliance consumption signals that summarize the different load behaviors corresponding to different programs. To do so, a clustering process is implemented on the activation events extracted from a training set of single appliance consumption signals completely disjoints in time from the test set. Activation event are aligned to all start at $t = 2$ and approximated using OMP. As the activation events are time series of different length, the standard Euclidean distance cannot be used to cluster them. it is used to measure distance between time series that may be shifted in time but it is also convenient to measure distance between time series of different length [22]. Clustering using a Dynamic time Warping (DTW) based distance was used successfully to cluster load profiles to align patterns that may be shifted in time [23]. DTW is a one-to-many points distance metric where each point of one time series is compared to many of the second time series [24]. It then form a matrix and the shortest path from the bottom left corner to the top right corner of the matrix is called warping path [25]. The sum of the Euclidean distances on the path is the DTW distance. The main benefits of DTW compared to Euclidean is to compensate for temporal translations in the patterns and it can also calculate distance between activation events of different durations (length of the time series). The pairwise DTW distance matrix of all the activation events is computed for each appliance a hierarchical ascending clustering using the Ward criterion is computed on the DTW distance matrix. The dendrogram of the hierarchical ascending clustering is used to generate the best partition into $3 \leq K \leq 6$ clusters which centroids form the representative signatures.

The identification of activation events from the communities is then done by calculating the DTW distances with the set of representative signatures of each appliances. The label of the closest representative signature is then assigned to the activation event. The labelling is a mapping process between labels and the shape of the signature thus it needs *prior* information about how the signature of the appliances of interest looks like. In other words, it cannot be done in a totally unsupervised manner.

III. Case Study

The algorithm performance is evaluated on the UK-Dale dataset [9] and benchmarked against the results obtained by [26] and [4] using respectively Factorial Hidden Markov Model (FHMM) and a Recurrent Neural Network (RNN) with a Long Short Term Memory (LSTM).

A first implementation with the same dataset at 6 seconds resolution is done to compare our results to the ones in [4], [26]. The consumption from one household over three months with five single consumption appliances: dishwasher, washing machine, kettle, microwave and fridge. This implementation is detailed in Section III-A.

A second implementation is performed on the same data at one minute resolution to evaluate how the performances of the algorithm evolves with the change of resolution.

A. Implementation

The overcomplete dictionary stores all the possible boxcar functions based on the combination of translation l and width w. Hence the size of the dictionary increases exponentially with the length of the signal to approximate. In practice, it is then not realistic to implement the OMP algorithm on large portions of data (i.e. days), thus our consumption signal has to be split into smaller batches of data. We fixed the maximum

length of a batch to 1000 time steps, our dictionary consists of all the boxcar functions with length between one to 1000 and represents 7.4 GB. The way batches are formed influences the output of OMP as it approximates the signals independently over each batch. In our implementation, batches have variable durations as the cut of the consumption time series is made in periods of lower consumption (less than 70 Watts) which are materialized with vertical gray lines in Figure 5. For the entire evaluation periods, the operation time of batches ranges from 30 seconds to 1 hour 20 minutes, with an average of 40 minutes which is in accordance with the duration of the activation events of the selected appliances. Before performing

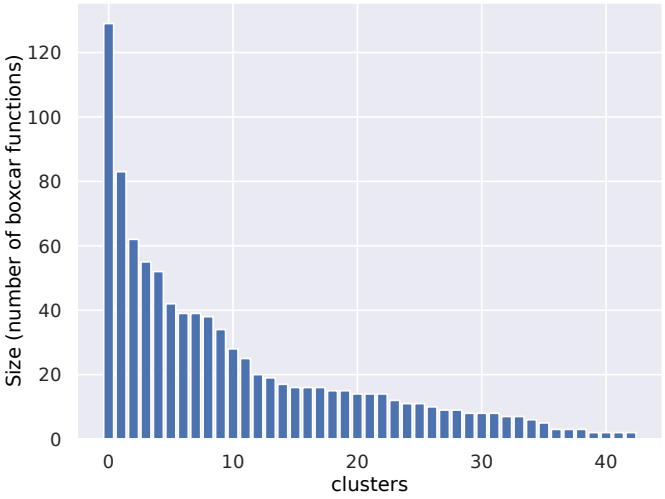

Fig. 3. Histogram of the clusters generated by the GMM.

any operation, a low pass filter sets power consumptions lower than 5 Watts to zero for all the datasets used for test, training or performance evaluation. The OMP has been applied using an error tolerance of 0.055 defined empirically. It eliminates high frequency variations and keeps only the large variations materializing the change of states of appliances as shown in red dashed line in Figure 5. For example the averaged RMSE generated by the approximation in Figure 5 is of 77 Watts, which is larger than the averaged RMSE over the entire period (53 Watts). The output of OMP, presented for the sample in the bottom of Figure 5, is the support vector Λ which consists of 6660 unique boxcar functions selected from the over-complete dictionary. The elements in Λ can have approximately the same amplitude α and width w but a different translation l (Figure 2). The boxcar functions are alike and we can suppose that they are produced by the same process. Hence we assume in this work that two boxcar functions which have similar shapes are likely to be generated by the same appliance.

A first filtering of the type-I appliances (e.g. fridge, microwave, kettle) is operated. The single boxcar functions which distance to a pattern from the labelling library (Figure 4) is under a defined threshold are labeled with the corresponding type-I appliances. The reason to execute the filtering at this stage is that type I appliances in this test have a high frequency of activation, especially the fridge, which means that they are often activated at the same time that larger appliances (e.g. washing machine or dishwasher). They remain unrelated

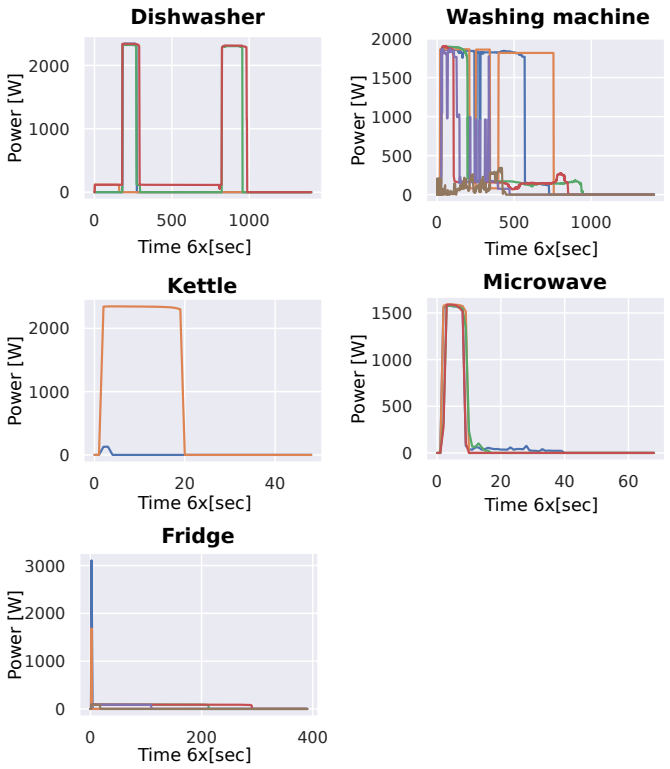

Fig. 4. Representative signature of each appliances

events but the community detection would gather them as a single activation event from the same appliance, forming a corrupted signature.

The GMM generates clusters of boxcar functions which have approximately the same shape $(\boldsymbol{\alpha}, \mathbf{w})$ and assumed to be generated by the same appliance (see Figure 1). The GMM uses only the number of clusters to be formed as parameter, which is set to 43 in the present work. Figure 3 shows the histogram of the cluster sizes.

The community detection requires two parameters, the modularity threshold set to 1 and the scaling parameter for the weights set to 0.95. The training set is sampled from the same household and corresponds to a year of data. The clustering process is run on all the activation events collected from the individual appliances consumption signals after sparse signal approximation. The hierarchical ascending clustering generates the following number of representative signatures; dishwasher: 5, washing machine: 6, microwave: 4, kettle: 2, fridge: 6. Figure 4 gives an overview of the representative signatures per appliance.

B. Performance Evaluation

The performance of our disaggregation algorithm is evaluated using both classification performance metrics and the estimation accuracy commonly used in the NILM literature [4], [27]. With classification algorithms, the performance of the prediction is evaluated in comparison to the ground truth. The recall or True Positive Rate (TPR), the precision, also known as Positive Predictive Value (PPV), and the accuracy (ACC) are defined as

$$\text{TPR} = \frac{\text{TP}}{\text{TP + FN}}, \quad \text{PPV} = \frac{\text{TP}}{\text{TP + FP}}, \quad \text{ACC} = \frac{\text{TP+TN}}{\text{P + N}}$$

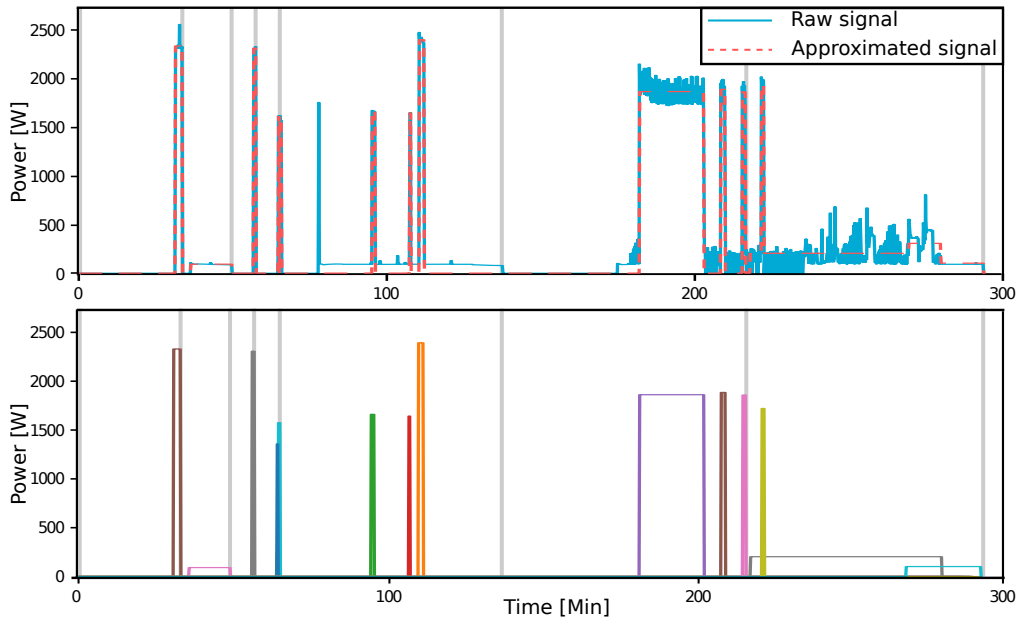

Fig. 5. Top: Raw consumption signal (blue) and the OMP approximated signal (red dashed). Bottom: detailed boxcar functions used to approximate the signal. the vertical gray lines correspond to the limits of the batches.

and are calculated from the output of the confusion matrix,

		Actual: Positive	Actual: Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)
		P	N

where True Positive/Negative (TP/TN) represents the number of times a disaggregated signal from a single appliance being ON/OFF is correctly assigned, False Positive (FP) represents the number of times a disaggregated signal from a single appliance is considered ON (consumption significantly larger than zero) but was actually OFF and False Negative (FN) is a disaggregated signal from a single appliance considered as OFF (consumption close to zero) but was ON. From the Precision and the Recall, the F_1-score

$$F_1 = 2\,\frac{\text{PPV TPR}}{\text{PPV + TPR}} \tag{12}$$

can be calculated. In the NILM literature, the estimation accuracy (EC)

$$\text{EC}^n = 1 - \frac{\sum_{t=1}^{T} |y_n^t - \hat{y}_n^t|}{2\sum_{t=1}^{T} y_n^t}, \tag{13}$$

where t is the time index and n is the appliance index, is used to estimate how accurately the consumption from each appliance n has been estimated [4], [6], [27]–[29]. The overall estimation accuracy (EC) is given by

$$\text{EC} = 1 - \frac{\sum_{t=1}^{T}\sum_{n=1}^{N} |y_n^t - \hat{y}_n^t|}{2\sum_{t=1}^{T}\sum_{n=1}^{N} y_n^t}. \tag{14}$$

and tells how much of the overall signal is kept after summing up the disaggregated signals. All the introduced performance evaluation measures are taking values in $[0, 1]$ and are read the same way: the higher the value, the higher the performance.

IV. Results and Discussions

The results of the implementation at six seconds are first evaluated against the benchmarks. Afterward, the evolution of the performance with the degradation of the data resolution is reported.

A. Benchmarking Against State-of-the-art Methodologies at High Resolution

The results of our methodology based on OMP are benchmarked against the results of FHMM as implemented in [26] and the RNN with LSTM, simply noted LSTM afterward, as implemented in [4] (Table I). In the first column 'Across Appliances', the only performance metric that is calculated across appliances is the estimation accuracy, the other performance metrics are simply the average across all appliances.

Looking first at the performance evaluation across appliances, the OMP approach has a slightly better EC that the FHMM and the LSTM. Regarding the classification performance, our approach outperforms the benchmarks for all of them besides the TPR. For the TPR our algorithm (0.55) performs slightly better than the FHMM (0.53) but is far from the LSTM (0.85). LSTM detects most of the activation events (high TPR) but have also a high false positive rate (lower PPV) compared to OMP.

The reading of the detailed performances of each appliances unveils large variability. It appears that the dishwasher is a difficult appliance to detect accurately, as all methods perform poorly. The OMP algorithm detects it less often than the FHMM and LSTM (lower TPR) but makes less false positives,

TABLE I
DISAGGREGATION PERFORMANCE OF THE OMP-GMM-COMMUNITY DETECTION APPROACH COMPARED TO FHMM [26] AND RNN-LONG SHORT TERM MEMORY (LSTM) [4].

Metrics	Accross Appliances			Dishwasher			Washing machine			Kettle			Microwave			Fridge		
Methods	OMP	FHMM	LSTM	OMP	FHMM	LSTM	OMP	FHMM	LSTM	OMP	FHMM	LSTM	OMP	FHMM	LSTM	OMP	FHMM	LSTM
EC	**0.93**	0.91	0.92	0.26	**0.91**	0.86	0.71	0.84	**0.88**	0.56	0.92	**0.99**	0.67	0.84	**0.98**	0.81	0.94	**0.97**
F_1-score	**0.54**	0.18	0.38	**0.26**	0.05	0.08	**0.61**	0.08	0.03	0.47	0.19	**0.93**	**0.54**	0.01	0.13	**0.83**	0.55	0.74
PPV	**0.62**	0.12	0.36	**0.24**	0.03	0.04	**0.73**	0.04	0.01	0.33	0.14	**0.96**	**0.79**	0.01	0.07	**0.99**	0.40	0.72
TPR	0.55	0.53	**0.85**	0.29	0.49	**0.87**	0.53	0.64	**0.73**	0.82	0.29	**0.91**	0.41	0.34	**0.99**	0.71	**0.86**	0.77
ACC	**0.96**	0.70	0.66	**0.96**	0.33	0.30	**0.96**	0.79	0.23	0.99	0.99	**1.00**	**0.99**	0.91	0.98	**0.88**	0.50	0.81

considering it is ON when it is actually OFF (higher PPV), which leads to a higher F_1-score for the OMP. For the washing machine, the EC of the OMP is higher than for the dishwasher. Against the benchmarks, its TPR is lower but its PPV is higher which generates again a better F_1-score for the OMP (0.61). For the kettle, the LSTM is almost perfect as it displays high performance for all metrics. Nevertheless the OMP performs better than the FHMM for the activation events detection (higher F_1-score, PPV, TPR). For the microwave, the detection of activation events (TPR) of the OMP is higher than the FHMM but lower than the LSTM which detects almost all activation events (TPR=0.99). However in terms of F_1-score the performance of the FHMM is again better. For the fridge the performance of all the methodology is high, the FHMM has the highest TPR (0.86). The OMP and the LSTM are performing similarly (respectively 0.71 and 0.77). The OMP has again a lower rate of false positive which generates a higher F_1-score for OMP compare to the benchmarks.

The performances for the individual appliances differ largely depending on the complexity of the load behavior. As the OMP performs an approximation of the signal into a square signal (Figure 5), appliances which show complex transient load behaviors, will yield poorer results. For the same reason appliances displaying similar average power consumptions and operation times are hard to separate. Comparing supervised against unsupervised learning algorithms is not a fair comparison and it is not expected that our algorithm outperform the benchmarks but just reaches similar performance.

B. Evolution of The Performance With the Degradation of the Resolution

The state-of-the-art approaches require high resolution to identify signatures based on transient states which means that their performance drops really quickly with the degradation of the resolution. As the OMP does not rely on the transient states, it is expected to perform correctly at lower resolution. An implementation of the OMP algorithm was done on the same dataset at one minute resolution to compare the performance and evaluate how it evolves with a degradation of the resolution. one minute resolution, divide already the number of points by ten and brings us closer to what is implemented at large scale.

The performance metrics across appliances displays that the performance at six seconds is better for all performance criteria besides the PPV (Table II). Looking closely at the values, it appears that they are close for the EC and which means that the performance do not reduce much with the change of resolution. The PPV at one minute is higher than at 6 seconds and the TPR is higher at 6 seconds that at 1 minute which results into similar F_1-score performances.

The individual appliances performance metrics, are showing that the change of in performances between six seconds and one minute are again depending on the appliance and its load behavior. For the dishwasher, The performance at one minute resolution is actually better than at six seconds resolution. The EC is doubled which means that the recovery of the signal is better. For the detection the F_1-scores are equivalent, but there are no false positive at one minute (PPV=1), but the TPR is lower than at six seconds (0.15 against 0.29). For the washing machine, the implementation at six seconds outperforms the implementation at one minute for all the metrics. For the kettle, the implementation at one minute recover better the signal that the one at six seconds (larger EC). Regarding the classification metrics, the F_1-score is also better at one minute as it shows a high PPV. However, the TPR is better at six seconds. For the microwave, the performance of the implementation at six seconds is better for all metrics but the TPR where the implementation at one minute is slightly better. For the fridge, the performances are both high.

Between six seconds and one minute, the number of data points is divided by ten, yet the performance of the OMP is not much deteriorated by this change of resolution besides the performance for the washing machine and the microwave. nevertheless, state-of-the-art methodologies relying on signatures in transient states would have their performances seriously altered by this change of resolution.

V. CONCLUSIONS AND FUTURE WORKS

The method presented in this paper is an application oriented and unsupervised approach to NILM. Indeed the state-of-the-art NILM algorithms suffer from a lack of generalization to any household and ability to perform at lower resolutions. As the OMP approach is unsupervised, no training using individual appliances consumption signals is required before running the algorithm. Hence it generates no training bias. It comes at the cost of a lower performance in terms of estimated accuracy compare to state-of-the-art approaches. It is a generic tool which can be used for different purposes at different resolution: high resolution (appr. 1 minute) to provide feedback to customers on their consumption (as presented in this work), medium (5-15 minutes) in a DSM framework to separate the consumption of appliances that could provide some services by moving or decreasing its consumption and provide feedback to aggregators or DSOs. However the benefit

TABLE II
COMPARISON OF THE PERFORMANCES OF THE OMP APPROACH WITH DATA AT SIX SECONDS AND ONE MINUTE RESOLUTION

Metrics	Accross Appliances		Dishwasher		Washing machine		Kettle		Microwave		Fridge	
Resolution	6s	1min	6s	1min	6s	1min	6s	1min	6s	1min	6s	1min
EC	**0.93**	0.91	0.26	**0.57**	**0.71**	0.58	0.56	**0.73**	**0.67**	0.35	**0.81**	**0.81**
F_1-score	**0.54**	0.53	**0.26**	0.25	**0.61**	0.54	0.47	**0.60**	**0.54**	0.44	**0.83**	0.82
PPV	0.62	**0.78**	0.24	**1.00**	**0.73**	0.59	0.33	**0.89**	**0.79**	0.47	**0.99**	0.98
TPR	**0.55**	0.45	**0.29**	0.15	**0.53**	0.51	**0.82**	0.45	0.41	**0.42**	**0.71**	**0.71**
ACC	**0.96**	**0.96**	0.96	**0.98**	**0.96**	0.95	**0.99**	**0.99**	**1.00**	0.99	**0.88**	0.87

of disaggregating down to the smallest appliances has a limited impact compare to the resources needed to execute it. Hence it is pragmatic to focus only on the detection and/or disaggregation of specific appliances with large consumptions (e.g. PV, EV, heat pump if controllable) that provides flexibility to the grid.

Such an approach is then better suited for real-world implementation as it can be used at various resolution. Subsequently it is also less computationally expensive as it handles less data. It will also reduce ethical concerns raised by NILM algorithm that could spy on high resolution energy consumption unveiling human behaviors.

As an extension of this work could be to apply the methodology to the use case of heat pumps, EVs or electric heater for example. These appliances have potential to be provide service to the grid and represent an increasing part of the load.

ACKNOWLEDGMENT

The authors would like to thank the EUDP for funding through the EnergyLab Nordhavn project (EUDP 64015-0055).

REFERENCES

[1] K. C. Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? the case of electricity," *Energy Policy*, vol. 52, pp. 213–234, 2013.

[2] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[3] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Artificial Intelligence and Statistics*, 2012, pp. 1472–1482.

[4] J. Kelly and W. Knottenbelt, "Neural nilm: Deep neural networks applied to energy disaggregation," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. ACM, 2015, pp. 55–64.

[5] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, 2012.

[6] N. Henao, K. Agbossou, S. Kelouwani, Y. Dubé, and M. Fournier, "Approach in nonintrusive type i load monitoring using subtractive clustering," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 812–821, 2017.

[7] B. Zhao, L. Stankovic, and V. Stankovic, "Blind non-intrusive appliance load monitoring using graph-based signal processing," in *Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on*. IEEE, 2015, pp. 68–72.

[8] S. Arberet and A. Hutter, "Non-intrusive load curve disaggregation using sparse decomposition with a translation-invariant boxcar dictionary," in *Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2014 IEEE PES*. IEEE, 2014, pp. 1–6.

[9] J. Kelly and W. Knottenbelt, "UK-DALE: A dataset recording UK Domestic Appliance-Level Electricity demand and whole-house demand," *ArXiv*, vol. 59, 2014.

[10] A. Faustine, N. H. Mvungi, S. Kaijage, and K. Michael, "A Survey on Non-Intrusive Load Monitoring Methodies and Techniques for Energy Disaggregation Problem," *arXiv*, no. March, 2017.

[11] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[12] L. Rebollo-Neira and Z. Xu, "Sparse signal representation by adaptive non-uniform B-spline dictionaries on a compact interval," *Signal Processing*, vol. 90, no. 7, pp. 2308–2313, 2010.

[13] J. Xu, H. He, and H. Man, "Active Dictionary Learning in Sparse Representation Based Classification," *arXiv*, no. 1, pp. 1–7, 2014.

[14] L. Daudet, "Audio sparse decompositions in parallel," *IEEE Signal Processing Magazine*, vol. 27, no. 2, pp. 90–96, 2010.

[15] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar conference on signals, systems and computers*. IEEE, 1993, pp. 40–44.

[16] R. Gribonval, H. Rauhut, K. Schnass, and P. Vandergheynst, "Atoms of all channels, unite! average case analysis of multi-channel sparse recovery using greedy algorithms," *Journal of Fourier analysis and Applications*, vol. 14, no. 5-6, pp. 655–687, 2008.

[17] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[18] S. Fortunato, "Quality functions in community detection," in *Noise and Stochastics in Complex Systems and Finance*, vol. 6601. International Society for Optics and Photonics, 2007, p. 660108.

[19] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, oct 2008.

[20] X. Que, F. Checconi, F. Petrini, and J. A. Gunnels, "Scalable community detection with the louvain algorithm," in *Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International*. IEEE, 2015, pp. 28–37.

[21] M. E. J. Newman, "Analysis of weighted networks," *Phys. Rev. E*, vol. 70, no. 5, p. 056131, Nov. 2004.

[22] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering–a decade review," *Information Systems*, vol. 53, pp. 16–38, 2015.

[23] G. Le Ray and P. Pinson, "Online adaptive clustering algorithm for load profiling," *Sustainable Energy, Grids and Networks*, vol. 17, p. 100181, 2019.

[24] T. Warren Liao, "Clustering of time series data - A survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[25] J. Paparrizos and L. Gravano, "k-shape: Efficient and accurate clustering of time series," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 1855–1870.

[26] N. Batra, J. Kelly, O. Parson, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh, and M. Srivastava, "Nilmtk: an open source toolkit for non-intrusive load monitoring," in *Proceedings of the 5th international conference on Future energy systems*. ACM, 2014, pp. 265–276.

[27] S. Makonin and F. Popowich, "Nonintrusive load monitoring (NILM) performance evaluation: A unified approach for accuracy reporting," *Energy Efficiency*, vol. 8, no. 4, pp. 809–814, jul 2015.

[28] J. Z. Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA*, vol. 25. Citeseer, 2011, pp. 59–62.

[29] B. Zhao, L. Stankovic, and V. Stankovic, "On a Training-Less Solution for Non-Intrusive Appliance Load Monitoring Using Graph Signal Processing," *IEEE Access*, vol. 4, pp. 1784–1799, 2016.