

Supervised Energy Disaggregation using Dictionary - Based Modelling of Appliance States

C. Mavrokefalidis^{1,2}, D. Ampeliotis^{1,2}, E. Vlachos^{1,2}, K. Berberidis^{1,2}, and E. Varvarigos^{1,3}

¹Computer Technology Institute, 26500, Rio - Patras, Greece

²Department of Computer Engineering and Informatics, University of Patras, Greece

³Department of Electrical and Computer Engineering, National Technical University of Athens, Greece

E-mails: {maurokef, ampeliot, vlassose, berberid, manos}@ceid.upatras.gr

Abstract—In this paper, a supervised energy disaggregation method is proposed. The appliances to be monitored, are modelled by multi-state finite state machines. Each state of an appliance is described by exactly one vector of power consumptions from a carefully designed set of such vectors (called atoms), that comprise a dictionary. The latter is constructed during a training phase, where it is assumed that individual power consumption signals are available. A clustering algorithm is applied on overlapping patches extracted from the training signal to select a fixed number of patches, i.e., the atoms of the dictionary. Moreover, in the training phase, an appropriate state transition matrix is constructed. During the operation phase, where the actual disaggregation task is performed, a trellis, with a reduced number of transitions, is used for the acquisition of the disaggregated signals per appliance. Numerical results, using the REDD dataset, are provided, in order to demonstrate the effectiveness of the proposed method.

I. INTRODUCTION

Non-Intrusive Load Monitoring (NILM), also known as energy disaggregation, refers to the problem of analyzing an aggregate energy signal, such as the one coming from a whole-home power monitor (smart meter), and extracting information about different individual loads of the system [1]. In Fig. 1, we demonstrate this idea. In particular, we consider a smart meter that measures the total consumption (e.g. in terms of power, current, etc.) of four devices, i.e. a lamp, an oven, a computer screen and a fan. Based on this aggregate signal, NILM tries to answer questions such as what is the individual consumption of each device and to quantify the percentage of consumption imposed by each device.

The usefulness of NILM has been already identified by early works like [1]. For example, NILM can be used to generate energy consumption reports, that can aid consumers in following more energy efficient practices. Nowadays, the appearance of the smart-grid vision [2] has led many countries to start investing in the modernization of their power grids, including the vast installation of (central) smart meters in the (until recently passive) residential and commercial areas. Moreover, there are many studies suggesting that high energy efficiency and savings could be achieved if end-users became aware, in detail, about their energy consumption profiles and

This work was supported by EU funds via the FP7 VimSEN project (<http://www.ict-vimsen.eu/>)

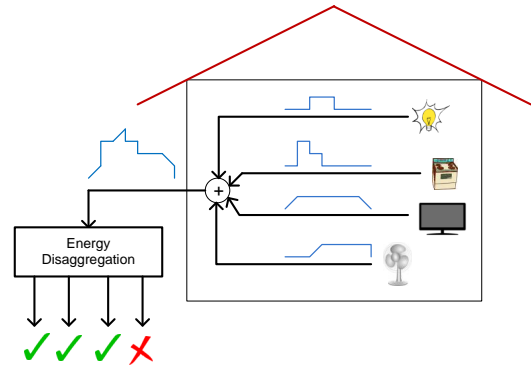


Fig. 1. At home, an aggregate signal from four devices, is used as input to the energy disaggregation block

received appropriate recommendations in real-time (e.g. [3]). For the above reasons, NILM techniques have received a renewed interest, and they are part of many current research and innovation actions [4], [5], [6].

The existing methods, for NILM, comprise two distinct tasks [7]; (a) a proper modeling of the individual appliances, by extracting informative features, based on their typical consumption behavior, and (b) derivation of an algorithm capable of deciding which devices are active at a particular time interval by utilizing information from an aggregate energy consumption signal. The process of constructing informative descriptions of the individual devices and, subsequently, disaggregating the available energy signal, is a learning problem that can utilize supervised or unsupervised methods. The former methods, following an off-line initial training phase, use known information about each device to be monitored. For example, in [8], a sparse coding approach is adopted to model each device by a set of “atom” signals which comprise a dictionary. A similar approach is followed in [9] where both the training and disaggregation phases are based on small patches of the involved signals. In [10], a modified Viterbi algorithm is proposed, for on-off devices, with complexity which, due to the inherent sparsity of the involved transition probability matrices, is linear with respect to the number of the devices. For multi-state devices, [11] proposed a sparse Viterbi algorithm that is able to disaggregate a larger number of devices. In [12], device harmonics are utilized and an $l1$ -

norm optimization problem is described for NILM, aiming at identifying whether the devices are on or off in the time interval of interest. In [13], a semi-supervised approach is proposed where a general model, for each device, is adopted and a training procedure is described for the inference of the model parameters from the available aggregate signal. In [14], an NILM technique that employs k-means-based clustering and classification (using support vector machines), is proposed. The training phase is based on either home-specific or home-agnostic device data. For the latter, a generic database with numerous device signatures is utilized.

Unsupervised methods, on the other hand, are applied directly on the aggregate signal and assume no prior knowledge about the devices. In [15], four variants of a Hidden Markov Model (HMM) are studied for modeling the data originating from low frequency sampling. In [16], an additive factorial HMM is adopted and, in particular, two models are considered; for the total aggregate output and for the difference between successive outputs. In [17], a classification of devices is proposed that is based on their “power consumption units” and working styles. Disaggregation of the signal is approached through an event detection scheme that uses clustering. In [18], a factorial HMM is adopted which is able to exploit information regarding device interactions due to their connection to a common underlying electric circuit.

In this paper, a novel approach for solving the disaggregation problem, is presented. The method relies upon works like [1] and [9] in the sense that it exploits properly some of their advantageous aspects. More specifically, we adopt a multi-state modeling, for each appliance, where each state is described by a vector of power consumptions measured at subsequent time instants (as in [9]), rather than a single measurement. However, different from [9], where a so-called sparse subset selection algorithm was employed for the training phase, we propose a training method that relies on clustering. Finally, in order to keep the computational complexity of the disaggregation algorithm to a tractable level, we propose to suitably prune the trellis - diagram which is computed during the training phase. Finally, the method is evaluated using power consumption signals from REDD [19].

In the following, first, the problem of energy disaggregation is described in Sec. II. Then, in Sec. III, the proposed technique is presented focusing on its training and normal operation phases. In Sec. IV, the evaluation of the technique is present where information is provided about the utilized dataset, implementation issues and, lastly, the simulation results. Finally, Sec. V concludes the paper.

Notation: $(\cdot)^T$ denotes the transpose of a matrix. Vectors and matrices are denoted with bold, small and capital letters, respectively.

II. ENERGY DISAGGREGATION

A. Problem Formulation

Let us consider a set of K (home) appliances and their respective power consumption signals $y_k[t]$, where k denotes the appliance index $k = 1, 2, \dots, K$ and $t = 1, 2, \dots, T$

represents discrete time. Furthermore, let us consider that the measured, noisy, aggregate power consumption signal can be written as,

$$y[t] = \sum_{k=1}^K y_k[t] + w[t], \quad (1)$$

where $w[t]$ denotes measurement noise and other inaccuracies of the model. The objective of the energy disaggregation problem is to estimate the energies E_k consumed by each one of the K devices, where

$$E_k = \sum_{t=1}^T y_k[t] \quad k = 1, 2, \dots, K, \quad (2)$$

using the measured aggregate power consumption signal $y[t]$ in the time interval $[1, T]$ and any available (a-priori) information regarding the operation of the K appliances. Usually, in the supervised version of the problem, the a-priori information may be in the form of a set of so-called training signals $y_k[t]$ for $t \in [t_1, t_2]$, from power measurement devices that were temporarily installed to monitor the consumptions of each individual appliance.

B. Relevant prior work

In order to better explain the proposed disaggregation approach and, furthermore, to demonstrate the differences of our method as compared to similar ones, we briefly mention here the key points of such existing methods. In [1], it was proposed to model each device as a finite state machine (FSM), where each state corresponds to a specific power consumption. Thus, given a set of models and an aggregate power consumption $y[t]$, the disaggregation problem at time t can be written as

$$\{s_1^{(o,t)}, s_2^{(o,t)}, \dots, s_K^{(o,t)}\} = \arg \min_{s_k \in S_k} \left| y[t] - \sum_{k=1}^K s_k \right|, \quad (3)$$

where S_k is the set of all states for device k , each element $s_k \in S_k$ is the respective power consumption and $s_k^{(o,t)}$ is the optimal state for device k at time t . This approach, known as the combinatorial optimization (CO) approach to the disaggregation problem, suffers from high computational complexity. In particular, given that $|S_k| = |S|$ for all devices, the complexity of the CO approach is $\mathcal{O}(|S|^K)$. Another drawback of the CO approach is that it does not, in general, manage to discriminate between devices with states that correspond to the same power consumptions, even though these devices may have very different “energy footprints” in time. Thus, approaches that utilize the time-structure of the power consumption signals of each device have appeared in the literature.

In [9], the proposed method consists of two phases; the first one for training and the second one for the disaggregation task itself. One key point, in both phases, is that the adopted procedures rely upon small patches of the available signals which leads to an online, real-time operation for the latter phase. In more detail, in the training phase, it is assumed that a consumption signal per device is available. From each

signal, patches (of length w) are extracted and they are used for designing a corresponding dictionary that comprises a small set of carefully selected patches, which capture the dynamics (or different states) of the device. This set is determined based on a sparse subset selection algorithm that selects the most representative patches, which are capable of describing the device consumption signal. The main drawback of this approach is its high complexity (both in terms of computations and storage). When the dictionaries are determined, the disaggregation phase is executed, again, on patches of the aggregate signal. A minimization problem is defined based on a cost function that consists of an l_1 -norm representation error term and a regularization term that incorporates prior information about the devices (e.g. the concurrent operation of two devices). The solution selects one and only one patch from the dictionary of each device. This fact provides a connection to clustering-based dictionary learning methods [20] that employ the simpler k -means algorithm during the training phase.

III. THE PROPOSED APPROACH

In this section, the proposed method is presented. First, the training procedure is described and, then, the disaggregation approach that is adopted, is explained.

A. Training phase

The scope of the training phase is to compute a model that explains the power consumption signal of a device. In particular, we assume that the operation of the appliances that are of interest to us, can be modeled by an FSM. Thus, our scope is to compute a state diagram for each of the K devices, given the “training” signals $y_k[t]$, for $t \in [t_1, t_2]$ that we call the training time interval. In contrast to [1], where a single power consumption was used to define each state, we let each state correspond to a vector of w consecutive power consumptions, and we call this vector an atom. This is similar to the work in [9] where a dictionary is used to represent the states of a device. This approach makes it possible to discriminate among devices with states of similar powers, as long as these devices have different behavior in intervals of length w . Of course, considering that even consecutive signal patches of length w of the signals $y_k[t]$ will, in general, vary significantly, then an increased number of states will be required to accurately model these signals.

The first step, in the proposed learning procedure for device k , is to use the signal $y_k[t]$ to construct a matrix

$$\mathbf{Y}_k = [y_k[t_1] \ y_k[t_1 + 1] \ \cdots \ y_k[t_2 - w + 1]] \ , \quad (4)$$

where the vectors $\mathbf{y}_k[t]$ are defined as

$$\mathbf{y}_k[t] = [y_k[t] \ y_k[t + 1] \ \cdots \ y_k[t + w - 1]]^T \ . \quad (5)$$

In other words, matrix $\mathbf{Y}_k \in \mathbb{R}^{w \times (L-w+1)}$, $L = t_2 - t_1 + 1$ contains all the possible patches of length w that we can extract from the signal $y_k[t]$ for $t \in [t_1, t_2]$.

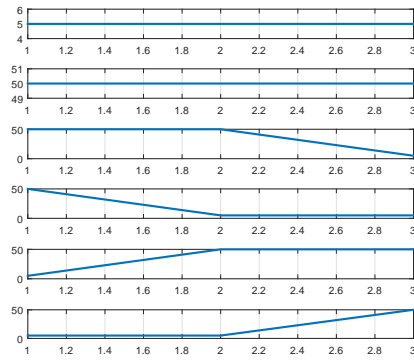


Fig. 2. Example atoms for a simple device that consumes 5 Watts at standby, 50 Watts during operation and when it operates it stays at this state for exactly 3 time instants

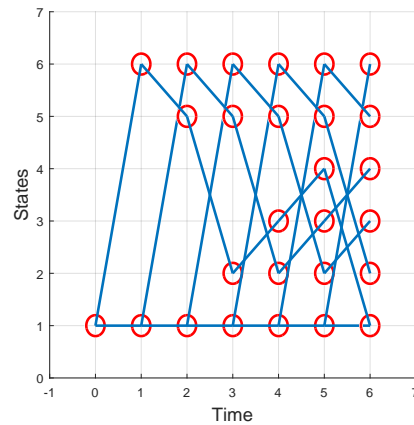


Fig. 3. Example trellis diagram for the device of Fig. 2

In the sequel, we employ some clustering algorithm (e.g., the k -means algorithm) to compute N representative patches. In particular, we compute

$$\{\mathbf{D}_k, \mathbf{c}_k\} = \text{Cluster}(\mathbf{Y}_k, N) \ , \quad (6)$$

where $\mathbf{D}_k \in \mathbb{R}^{w \times N}$ is the matrix of centroids/representatives and \mathbf{c}_k is a vector with $L - w + 1$ elements, defined as

$$\mathbf{c}_k = [c_k[t_1] \ c_k[t_1 + 1] \ \cdots \ c_k[t_2 - w + 1]]^T \ , \quad (7)$$

and $c_k[t] \in \{1, 2, \dots, N\}$ is the index of the corresponding centroid/representative for patch $\mathbf{y}_k[t]$.

Having computed matrix \mathbf{D}_k , we have determined the atoms of the N states of the FSM. The required transition probabilities can be estimated in a matrix \mathbf{P}_k , with entries

$$[p_k]_{i,j} = \frac{\sum_{t=t_1}^{t_2-w} I(c_k[t] = i \ \text{AND} \ c_k[t+1] = j)}{L - w} \ , \quad (8)$$

with $I(\cdot)$ an indicator function that returns one if its argument is true and zero otherwise. Thus, the matrices \mathbf{D}_k and \mathbf{P}_k describe the finite state machine for appliance k .

To demonstrate the ideas of the training procedure described in the previous paragraph, we provide a simple example.

Consider a simple appliance that consumes 5 Watts at standby and 50 Watts during operation. The device can start operating at any time, but when it starts it will operate for exactly 3 time instants. Furthermore, device activations occur at time instants that are more than 5 time instants apart. Assuming that we measure the power consumption signal of this device in the absence of measurement noise, and by applying our training procedure, the resulting atoms (i.e., columns of matrix \mathbf{D}) are shown in Fig. 2. Also, the resulting trellis diagram, drawn with the assumption that the device starts at standby, is shown in Fig. 3. Note that even though the device has only two distinct power consumptions, 6 states are required to accurately model this device when $w = 3$. Also, it is interesting to note that the resulting trellis diagram is sparse, in the sense that each of the 6 states is followed by a limited number of states.

B. Operation phase

After the training phase has been completed for all K devices, we can construct a super-trellis described by a matrix $\mathbf{D} \in \mathbb{R}^{w \times N^K}$ with columns all the possible sums of atoms, where in each sum exactly one atom per device (i.e., one column from each \mathbf{D}_k) is used, and the associated matrix $\mathbf{P} \in \mathbb{R}^{N^K \times N^K}$ that holds the super-state transition probabilities. This approach is equivalent to the so-called factorial Markov model, in which a number of Markov models operate independently in parallel and we observe only the sum of their outputs, which corresponds to a super-state [16]. In this work, in order to reduce the computational complexity, we assume that at time t the correct super-state has been detected and thus, we only search for the next super-state by examining only transitions that begin from the current state. Furthermore, we suitably prune the individual trellis diagrams of each device, by keeping only the N' most probable transitions. Thus, the computational complexity of this scheme is $\mathcal{O}(N'^K)$, which is significantly smaller than $\mathcal{O}(N^K)$ when $N' \ll N$.

Remark: It is pointed out here that the particular algorithm, which is adopted for the execution of the operation phase, is actually independent of the training procedure that is followed. In essence, the particular supervised training, where a device-specific dictionary is determined, can also be utilized, as is, by a disaggregation algorithm like the one that is proposed in [9]. This remark is interesting because the same training procedure can be tailored for two fundamentally different approaches that are followed in the relevant literature.

IV. NUMERICAL RESULTS

A. Dataset for energy disaggregation

In recent years, a number of datasets that are tailored, among other, for the evaluation of NILM techniques, are publicly available to the research community. Two examples are the Reference Energy Disaggregation Data-set (REDD, [19]) and the Dutch Residential Energy Data-set (DRED, [21]). The interested user is referred to [21] for more examples.

In this work, the proposed disaggregation technique is evaluated on REDD [19]. In general, this data-set contains power consumption curves for the two power mains and individual

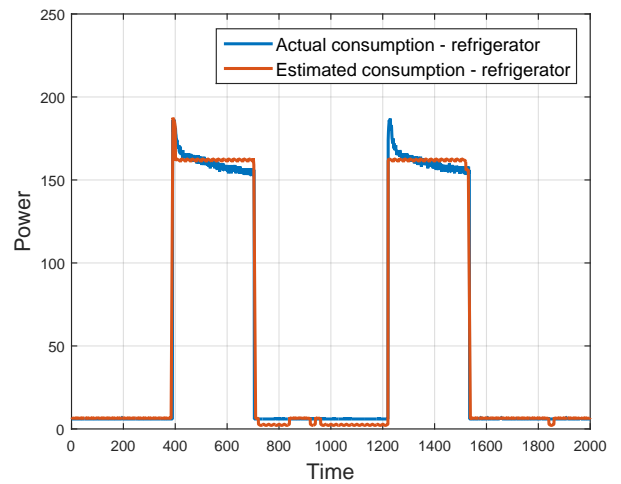


Fig. 4. Actual and reconstructed power consumption signals for the refrigerator

circuits, called channels, of six different houses. For the power mains signals, the sampling frequency is about one sample per second, while, for the individual circuits, the frequency is about one sample every 3 seconds. In Fig. 4, a segment of the power consumption signal of the channel “Refrigerator”, from the second house, is presented, along with its estimated reconstruction curve.

The consumption signals, for each channel, correspond to a number of days, as can be concluded from the UTC time stamps that were recorded along with the samples. For the evaluation needs in this work, two segments are extracted from each channel signal; called training and testing segments, respectively. The first one is used for the training phase, as dictated in Sec. III-A, while both are used for evaluating the performance of the proposed technique, as explained in Sec. III-B. It is noted here that the aggregate signal is assumed for simplicity to be produced as the sum of the corresponding segments (i.e. either the training or the testing ones) of all channels participating in the disaggregation task. Moreover, the channels are not separated into two groups (with respect to the measurement of the two mains in REDD). Thus, it is assumed that all channels are supported by the same main power supply of the house.

B. Implementation issues

In our experiments, we use the low-frequency dataset for “house 2” of the REDD database. This dataset contains power consumption measurements for 9 devices (thus, $K = 9$) sampled at a rate of one measurement in (about) every 3 seconds. We use the first seven days for training, that correspond to 151754 samples. We perform clustering on the matrices \mathbf{Y}_k for various values for the length w of the patches and for the number of centroids/representatives N . For clustering, we use the K-means algorithm with random initial centroids. As this approach is sensitive to initialization, we execute the algorithm

a number of times - each trial with a different initialization, and we use two metrics to select one of the resulting trellises:

- I The first metric that we used to select one of the outputs of the K-means algorithm was the minimum mean square error, in the sense that these centroids better represent the training patches. We call this approach Scheme I.
- II The second metric that we used takes into consideration the subsequent pruning of the trellis diagram. More specifically, for each trial we compute the corresponding matrix of transition probabilities, and in the sequel we compute the sum of the entries of this matrix that would be set to zero if pruning was applied. We select the output of the algorithm with the minimum such sum. We call this approach Scheme II.

After training, we use the following three days for testing (i.e. subsequent 68580). In our experiments, we prune the computed trellis diagrams of all the devices by considering only the $N' = 3$ most probable transitions from each state to the states of the next time instant. Thus, the disaggregation algorithm examines 3^9 super-states at each time instant (i.e., 3^9 columns of matrix \mathbf{D} , which has a total of N^9 columns, and in general $N > 3$), and selects the one that achieves the minimum squared error from the respective patch of the aggregate power signal.

C. Results

To evaluate the performance of the proposed method, we conducted some computer experiments. In particular, we examined the cases where w is set equal to 1, 5, 10 and 20. Also, we examined the case where $N = N' = 3$, i.e., no pruning is performed, and the case where $N = 10$ where pruning to $N' = 3$ transitions is performed. In this case, we examine both Scheme I and Scheme II. For all the above cases, we compute the disaggregation consumption metric, which is given by [8],

$$C = \frac{\sum_{k=1}^K \min(E_k, \hat{E}_k)}{\sum_{k=1}^K E_k}, \quad (9)$$

and the disaggregation accuracy metric, which is given by [9],

$$A = 1 - \frac{\sum_{t \in \{1, w+1, 2w+1, \dots\}} \sum_{k=1}^K \|\hat{\mathbf{y}}_k[t] - \mathbf{y}_k[t]\|_1}{2 \sum_{t \in \{1, w+1, 2w+1, \dots\}} \|\mathbf{y}[t]\|_1}, \quad (10)$$

where E_k is the actual energy consumed by device k at the examined interval, and \hat{E}_k is the estimated energy consumed by device k , where both energies are normalized to have the same sum. Moreover, $\mathbf{y}_k[t]$, $\hat{\mathbf{y}}_k[t]$ and $\mathbf{y}[t]$ are the actual and estimated patches for device k and the patch for the aggregated signal, respectively. Fig. 5 summarizes our results for the two performance metrics. The main observations that we can make, are summarized in the following:

- Scheme II manages to achieve better performance than Scheme I, for small values of w . For greater values of w , Scheme I obtains similar or better performance.

- The best disaggregation accuracy is about 60%. This is an expected results because, at the operation phase, a trellis-based approach is used that is similar to FHMMs and a similar outcome has been reported in [9].
- Although the best A -values are about 60%, the best C value is about 85% which provides a different view regarding the efficiency of the method. As reported in [22], [23], the task of disaggregation is highly related to the targeted application and the questions that need to be answered. For example, if the percentage of power consumption per device is desired then the C metric is more appropriate than the A metric that represents the accuracy of reconstructing the actual signal of a device.

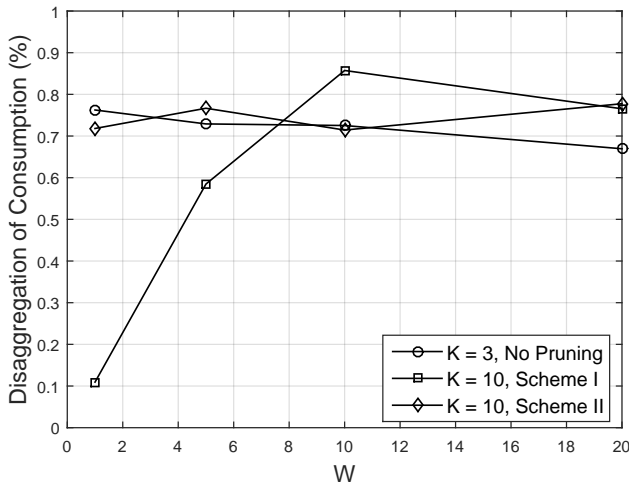
Finally in Fig. 6 we demonstrate the resulting energy percentages per device, as estimated by the proposed method for the cases with $w = 20$, as well as the actual energy consumptions. We note that for this specific experiment, the disaggregation problem is solved quite accurately.

V. CONCLUSIONS

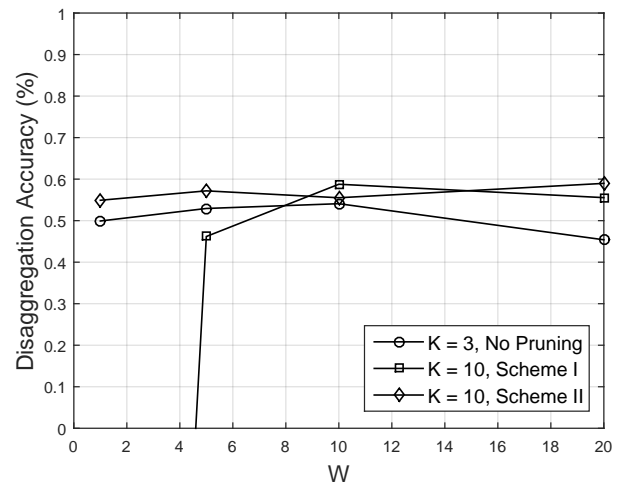
In this paper, a supervised disaggregation method was proposed that models the states of an appliance, using a dictionary designed by employing a clustering algorithm. A trellis-based algorithm is used for the disaggregation step, that operates on a properly pruned diagram, for complexity reasons. Using consumptions signals from REDD, the applicability of the proposed method was demonstrated. However, although the two schemes reduce the number of transitions, complexity still remains prohibitive for houses with more (than 9) appliances to be monitored (as is the case with other houses in REDD). Moreover, a better understanding of the intertwining of the key parameters of the method (i.e. N , N' , w) is required either by theoretical arguments or by appropriate simulated scenarios. The above two points dictate our future research efforts for this problem.

REFERENCES

- [1] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec 1992.
- [2] E. Commission, "Vision and strategy for europe's electricity networks of the future," *SmartGrids Technology Platform*, 2006.
- [3] K. Ehrhardt-Martinez, K. A. Donnelly, S. Laitner *et al.*, "Advanced metering initiatives and residential feedback programs: a meta-review for household electricity-saving opportunities." American Council for an Energy-Efficient Economy Washington, DC, USA, 2010.
- [4] 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.
- [5] 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.
- [6] A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. P. O'Hare, "Real-time recognition and profiling of appliances through a single electricity sensor," in *Proc. of the 7th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks (SECON)*, June 2010, pp. 1–9.
- [7] Y. F. Wong, Y. A. ekerciolu, T. Drummond, and V. S. Wong, "Recent approaches to non-intrusive load monitoring techniques in residential settings," in *Proc. of the IEEE Symposium on Computational Intelligence Applications in Smart grid (CIASG)*, April 2013, pp. 73–79.



(a) Consumption metric C of Eq. (9)



(b) Accuracy metric A of Eq. (10)

Fig. 5. Comparison of the “no-pruning” case ($N = 3$) with the two proposed schemes ($N = 10$ and $N' = 3$ versus the size w of the employed patches.

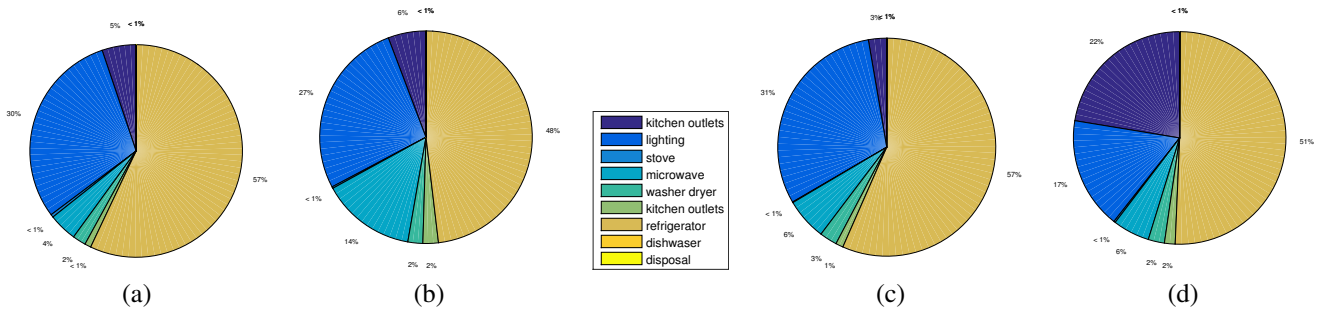


Fig. 6. (a) Actual energy consumptions, (b) Estimated energy consumptions for $N = 3, w = 20$, (c) Estimated energy consumptions for Scheme I, $N = 10, w = 20$, (d) Estimated energy consumptions for Scheme II, $N = 10, w = 20$

- [8] J. Z. Kolter, S. Batra, and A. Y. Ng, “Energy disaggregation via discriminative sparse coding,” in *Proc. of Advances in Neural Information Processing Systems*, 2010, pp. 1153–1161.
- [9] E. Elhamifar and S. Sastry, “Energy disaggregation via learning powerlets and sparse coding,” in *AAAI*, 2015, pp. 629–635.
- [10] M. Zeifman and K. Roth, “Viterbi algorithm with sparse transitions (VAST) for nonintrusive load monitoring,” in *Proc. of IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG)*, April 2011, pp. 1–8.
- [11] S. Makonin, F. Popowich, I. V. Baji, B. Gill, and L. Bartram, “Exploiting HMM sparsity to perform online real-time nonintrusive load monitoring,” *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–11, 2015.
- [12] J. T. Chiang, T. Zhang, B. Chen, and Y.-C. Hu, “Load disaggregation using harmonic analysis and regularized optimization,” in *Proc. of Asian-Pacific Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2012, pp. 1–4.
- [13] O. Parson, S. Ghosh, M. Weal, and A. Rogers, “Non-intrusive load monitoring using prior models of general appliance types,” in *Proc. of 26th AAAI Conference on Artificial Intelligence*, 2012.
- [14] H. Altrabalsi, V. Stankovic, J. Liao, and L. Stankovic, “Low-complexity energy disaggregation using appliance load modelling,” *AIMS Energy*, vol. 4, no. 1, pp. 1–21, 2015.
- [15] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, “Unsupervised disaggregation of low frequency power measurements,” in *Proc. of the SIAM International Conference on Data Mining*, 2011.
- [16] J. Z. Kolter and T. Jaakkola, “Approximate inference in additive factorial hmms with application to energy disaggregation,” in *Proc. of International Conference on Artificial Intelligence and Statistics*, 2012, pp. 1472–1482.
- [17] Z. Wang and G. Zheng, “Residential appliances identification and monitoring by a nonintrusive method,” *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 80–92, March 2012.
- [18] M. Aiad and P. H. Lee, “Unsupervised approach for load disaggregation with devices interactions,” *Energy and Buildings*, vol. 116, pp. 96–103, 2016.
- [19] J. Z. Kolter and M. J. Johnson, “REDD: A public data set for energy disaggregation research,” in *Proc. of Workshop on Data Mining Applications in Sustainability (SIGKDD)*, vol. 25. Citeseer, 2011, pp. 59–62.
- [20] I. Tosic and P. Frossard, “Dictionary learning,” *Signal Processing Magazine, IEEE*, vol. 28, no. 2, pp. 27–38, March 2011.
- [21] A. S. Uttama Nambi, A. Reyes Lua, and V. R. Prasad, “LocED: Location-aware energy disaggregation framework,” in *Proc. of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, 2015, pp. 45–54.
- [22] N. Batra, R. Baijal, A. Singh, and K. Whitehouse, “How good is good enough? re-evaluating the bar for energy disaggregation,” *CoRR*, vol. abs/1510.08713, 2015. [Online]. Available: <http://arxiv.org/abs/1510.08713>
- [23] S. Barker, S. Kalra, D. Irwin, and P. Shenoy, “NILM redux: The case for emphasizing applications over accuracy,” in *Proc. of the International Workshop on Non-Intrusive Load Monitoring (NILM)*, June 2014, pp. 1–4.