Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 51-15 (2018) 275–280

Jump model learning and filtering for energy end-use disaggregation

V. Breschi * **D. Piga** ** **A. Bemporad** ***

* *Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy (e-mail: valentina.breschi@polimi.it).*
** *IDSIA Dalle Molle Institute for Artificial Intelligence SUSPI-USI, 6928 Manno, Switzerland (e-mail: dario.piga@supsi.ch)*
*** *IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100 Lucca, Italy (e-mail: alberto.bemporad@imtlucca.it).*

Abstract: Energy disaggregation aims at reconstructing the power consumed by each electric appliance available in a household from the aggregate power readings collected by a single-point smart meter. With the ultimate goal of fully automatizing this procedure, we first estimate a set of jump models, each of them describing the consumption behaviour of each electric appliance. By representing the total power consumed at the household level as the sum of the outputs of the estimated jump models, a filtering algorithm, based on dynamic programming, is then employed to reconstruct, in an iterative way, the power consumption at an individual appliance level.

Keywords: Jump models, Filtering, Energy disaggregation, Non-intrusive appliance load monitoring, Recursive estimate.

1. INTRODUCTION

A detailed knowledge of timings, peak-hours, and frequencies of use of electric devices in a household is a useful starting point to understand consumers' behaviours, identify consumption anomalies, design and assess the impact of energy efficiency programs and customized demand management strategies. Without meter readings at individual appliance level, *energy disaggregation* algorithms can be used to retrieve information on the consumption of the single appliances from aggregate power readings collected by a single-point smart meter. This solution leads to a reduction in the number of smart meters and smart plugs installed in a house, thus reducing costs for the installation, maintenance and replacement of the monitoring system.

The problem of energy disaggregation, also known in the literature as *Non-Intrusive Load Monitoring* (NILM), was addressed for the first time in (Hart, 1992). Since the seminal work of Hart, several algorithms for energy disaggregation have been developed (see, *e.g.*, (Esa et al., 2016; Faustine et al., 2017) for a detailed review on the topic). Among the existing disaggregation algorithms, we mention the optimization-based method in (Suzuki et al., 2008), which formulates the energy disaggregation problem in terms of integer programming, and the work in (Piga et al., 2016) which minimizes a convex cost function accounting for the fitting error and penalizing the time switch in the operating modes of the single appliances. Instead, the methods in (Kolter and Jaakkola, 2012; Cominola et al., 2017) are based on *Factorial Hidden Markov Models* (FHMMs). In these algorithms, the consumption behaviour of the single appliance is modelled by a *Hidden Markov Model* (HMM), with hidden states representing the operating conditions of the considered device.

This paper presents a novel iterative algorithm for non-intrusive load monitoring, which follows a similar rationale as the FHHM-based approaches. The behaviour of each appliance is modelled by a jump linear model, which is estimated from the appliance's signature through the coordinate-descent optimization algorithm recently proposed in (Bemporad et al., 2017), under the hypothesis that the appliances rarely change their operating mode over two consecutive time samples. Similarly to NILM methods based on HMMs, each state of the jump model is associated to an operating condition of the single appliance. Once jump models for each appliance are estimated, the disaggregation problem is formulated in terms of discrete-state filtering, and solved iteratively through *dynamic programming*. Overall, the iterative nature of the algorithm makes it particularly suited also for a real-time implementation.

The paper is organized as follows. After introducing the energy disaggregation problem in Section 2, the estimation of jump models describing the behaviour of each appliance is discussed in Section 3, while the disaggregation algorithm is presented in Section 4. The proposed method is tested against the AMPds dataset (Makonin et al., 2016), containing real energy consumption readings of a single house located in Canada. The obtained results are discussed in Section 5. Concluding remarks and directions for future works are given in Section 6.

⋆ This work was partially supported by the H2020-SPIRE-636834 project "DISIRE - Distributed In-Situ Sensors Integrated into Raw Material and Energy Feedstock", and by the H2020-723248 project "DAEDALUS - Distributed control and simulation platform to support an ecosystem of digital automation developers".

10.1016/j.ifacol.2018.09.147

2. PROBLEM FORMULATION

Suppose that N different devices connected to the power line are available in a house and let $y_i(t)$, with $i \in \{1,\dots,N\}$, be the power demand of the i-th appliance at time t.

The aggregate power reading $y(t)$ measured at the household level is then given by the sum of the power demands of the single appliances, *i.e.*,

$$y(t) = \sum_{i=1}^{N} y_i(t) + e(t), \tag{1}$$

where $e(t)$ is a modelling error, accounting for additional unmodelled appliances available in the house and measurement noise on the aggregate reading.

In this work we want to address a *disaggregation problem*. Specifically, given a sequence $\mathcal{D}_T = \{y(t)\}_{t=1}^{T}$ of observations of the aggregate power signal, we aim at reconstructing the unknown power demand $y_i(t)$ of the single appliances at each time sample t.

3. LEARNING SINGLE-APPLIANCE MODELS

With the ultimate goal of retrieving the end-use consumption patterns $y_i(t)$ from the aggregate measure $y(t)$, models describing the behaviour of each single device are first estimated. To this aim, we approximate the power consumption of the i-th appliance at time t with a static hybrid model, which consists of a collection of $K_i \in \mathbb{N}$ static local models, each describing the power demand of the considered appliance at a given operating condition or *mode*, *i.e.*,

$$\hat{y}_i(t) = \theta_i^{s_i(t)}, \tag{2}$$

where $s_i(t) \in \mathcal{S}_i = \{1,\dots,K_i\}$ denotes the active mode at time t and the parameter $\theta_i^{s_i(t)} \in \mathbb{R}$ thus represents the power consumed by the i-th appliance at mode $s_i(t)$.

For each appliance, the parameters θ_i^j, with $j \in \mathcal{S}_i$, are estimated from a training set $\mathcal{M}_i$ which only consists of the power consumed by the i-th appliance gathered over an intrusive period of length $\bar{T}$. To reduce intrusiveness and costs, $\bar{T}$ should be as short as possible and it might be different for each appliance. Obviously, the sets $\mathcal{M}_i$ should be disjoint from the dataset $\mathcal{D}_T = \{y(t)\}_{t=1}^{T}$ where final disaggregation should be performed. The number K_i of local sub-models is supposed to be known a priori. Nonetheless, K_i can be selected by cross-validation, with an upper-bound on K_i dictated by the maximum tolerated complexity of the single-appliance model.

Learning appliance models (2) thus amounts to estimate the parameters $\Theta_i = (\theta_i^1,\dots,\theta_i^{K_i})$. This requires to reconstruct the sequences of active modes $S_i = \{s_i(t)\}_{t=1}^{\bar{T}}$, which is not directly observable from data.

According to the jump model fitting algorithm proposed in (Bemporad et al., 2017), Θ_i and S_i are estimated from data by minimizing the *loss function*

$$J_i(\Theta_i, S_i) = \sum_{t=1}^{\bar{T}-1} \left[\ell\left(y_i(t), s_i(t), \Theta_i\right) + \mathcal{L}_i(s_i(t), s_i(t+1))\right] + \ell\left(y_i(\bar{T}), s_i(\bar{T}), \Theta_i\right) \tag{3}$$

The function $\ell\left(y_i(t), s_i(t), \Theta_i\right)$ accounts for the *fitting error*, as it penalizes the mismatch between the measured and the model output. Among possible fitting costs, the following quadratic cost is used in this work:

$$\ell\left(y_i(t), s_i(t), \Theta_i\right) = \frac{1}{\bar{T}}\left(y_i(t) - \theta_i^{s_i(t)}\right)^2. \tag{4}$$

The term $\mathcal{L}_i(s_i(t), s_i(t+1))$ in (3) is introduced to penalize changes of discrete state, in accordance with the prior hypothesis that the operating mode $s_i(t)$ changes rarely over time, so that $s_i(t) = s_i(t-1)$ for most of the time instants $t = 2, 3, \dots$. This is a reasonable assumption when the power readings are taken at high-time resolution (*e.g.*, 1 min). This leads to the following choice of $\mathcal{L}_i(s_i(t), s_i(t+1))$:

$$\mathcal{L}_i(s_i(t), s_i(t+1)) = \lambda_i(s_i(t))\mathbf{1}(s_i(t+1) \neq s_i(t))$$

where $\mathbf{1}(s_i(t+1) \neq s_i(t))$ is the indicator function of the condition $s_i(t+1) \neq s_i(t)$.The parameters $\lambda_i(j)$, $j = 1,\dots,K_i$, can be tuned via cross-validation or they can be chosen by following the guidelines provided in (Bemporad et al., 2017). In the considered application, we impose $\lambda_i(j) = \lambda_i$ for all $j \in \mathcal{S}_i$ and then we select λ_i through cross-validation. Based on the active mode sequence resulting from the training procedure it is then possible to update $\lambda_i(j)$ as the empirical probability (with Laplace smoothing) of the i-th appliance to be in the same mode $j \in \mathcal{S}_i$ for two consecutive steps, *i.e.*,

$$\lambda_i(j) = \frac{\sum_{t=1}^{\bar{T}-1} \mathbf{1}(s_i(t) = j, s_i(t+1) = j) + 1}{\sum_{t=1}^{\bar{T}-1} \mathbf{1}(s_i(t) = j) + K_i^2}, \tag{5}$$

with $\mathbf{1}(s_i(t) = j, s_i(t+1) = j)$ and $\mathbf{1}(s_i(t) = j)$ being the indicator functions of conditions $\{s_i(t) = j$ and $s_i(t+1) = j\}$ and $s_i(t) = j$, respectively. If the underlying system satisfies the prior assumption on the discrete state, the computed values of $\lambda_i(j)$, with $j = 1,\dots,K_i$ and $i = 1,\dots,N$, are expected to be relatively high ($\lambda_i(j) \geq 0.9$).

Using the coordinate descent approach presented in (Bemporad et al., 2017) and outlined in Algorithm 1, the cost (3) is alternately minimized with respect to the parameters Θ_i (for a fixed S_i), and with respect to the mode sequence S_i (for fixed Θ_i). When the state sequence S_i is fixed, the minimization of (3) over Θ_i is performed analytically by least squares (Step 1.1). At Step 1.2, for fixed $\Theta_i = \Theta_i^h$, the cost $J_i(\Theta_i^h, S_i)$ in (3) is minimized through *dynamic programming* (DP) as follows.

We first compute the "terminal cost":

$$V_i(j, \bar{T}) = \ell(y_i(\bar{T}), j, \Theta_i^h) \tag{6}$$

for all $j \in \{1,\dots,K_i\}$. Then, for all $t = \bar{T},\dots,2$, the cost $V_i(k, t-1)$ of the i-th appliance being at state j at time $t-1$ is computed going backward from time $\bar{T}$ by following the shortest path, *i.e.*,

$$V_i(k, t-1) = \ell(y_i(t-1), k, \Theta_i^h) + \min_{j \in \mathcal{S}_i}(V_i(j, t) + \mathcal{L}_i(j)), \tag{7}$$

for all $k \in \mathcal{S}_i$, with $\mathcal{L}_i(j) = \lambda_i(j)\mathbf{1}(s_i(t+1) \neq s_i(t))$. The optimal sequence of active modes S_i^h is thus obtained going forward through the computed shortest path.

The computation of the state sequence $\mathcal{S}_i$ thus requires $O(\bar{T}K_i^2)$ operations (see equation (7)).

Algorithm 1 Learning i-th appliance jump model

Input: Training data set $\mathcal{M}_i = \{y_i(t)\}_{t=1}^{\bar{T}}$; number K_i of sub-models; initial mode sequence $S_i^0 = \{s_i^0(1), \dots, s_i^0(\bar{T})\}$; parameters $\lambda_i(j)$, $j = 1, \dots, K_i$; maximum number of iterations $h_{\max}$.

1. **iterate for** $h = 1, \dots$
   1.1. $\Theta_i^h \leftarrow \arg\min_{\Theta_i} \sum_{t=1}^{\bar{T}} \ell\left(y_i(t), s_i^{h-1}(t), \Theta_i\right)$;
   1.2. $S_i^h \leftarrow \arg\min_{S_i} J_i(\Theta_i^h, S_i)$;
2. **until** $S_i^h = S_i^{h-1}$ or $h = h_{\max}$.

Output: Estimated model parameters $\Theta_i^\star = \Theta_i^k$ and mode sequences $S_i^\star = S_i^k$.

4. ENERGY DISAGGREGATION

Once the models for each single appliance are estimated, the aggregate power readings in the set $\mathcal{D}_T$ should be split into end-use energy consumptions. This requires to reconstruct the active mode $s_i(t)$ for each appliance at each time instant only from the aggregate power readings $y(t)$ and the sub-model parameters Θ_i.

Let us introduce the *joint active mode* $s(t) \in \mathbb{N}^N$, with $s(t)$ being the collection of the appliances' modes at time t, *i.e.*, $s(t) = [s_1(t), \ \dots, \ s_N(t)]$. We indicate as $\mathcal{S}$ the set of all possible combinations taken by $s(t)$.

Energy disaggregation can be formulated as a discrete-state filtering problem and solved iteratively, according to (Bemporad et al., 2017), by minimizing the cost $J(S^t)$

$$J(S^t) = \eth(y(1), s(1)) + \sum_{\tau=2}^{t} \left[\eth(y(\tau), s(\tau)) + \mathcal{L}(s(\tau-1), s(\tau))\right], \tag{8}$$

$$\mathcal{L}(s(\tau-1), s(\tau)) = \lambda(s(\tau-1))\mathbf{1}(s(\tau) \neq s(\tau-1)) \tag{9}$$

with respect to S^t, where $S^t = \{s(1), \dots, s(t)\}$ is the active joint mode sequence up to time t. Similarly to (3), the cost in (8) both penalizes the fitting error on the aggregate power measurement $y(t)$ and the switch of the joint mode. In particular, the *fitting cost* is:

$$\eth(y(t), s(t)) = \left(y(t) - \sum_{i=1}^{N} \theta_i^{s_i(t)}\right)^2,$$

while the penalization term $\mathcal{L}(s(\tau-1), s(\tau))$ in (9) accounts for the hypothesis that all the appliances rarely change their operating regime over time. Assuming that the mode transitions of the different devices are independent, the parameter $\lambda(d)$ in (8), with $d \in \mathcal{S}$, is chosen to be equal to

$$\lambda(d) = w \prod_{i=1}^{N} \lambda_i(d_i), \ w > 0, \tag{10}$$

where w is a tunable weight and $\lambda_i(d_i)$ is the empirical probability that the i-th appliance remains at the mode d_i for two consecutive time instants and it can be computed from the training datasets $\mathcal{M}_i$ as in (5).

The cost $J(S^t)$ in (8) is minimized iteratively through dynamic programming as described in Algorithm 2. At Step 1, the "initial cost" $\mathcal{J}(1, h)$ is computed for all $h \in \mathcal{S}$ and the joint mode at time $t = 1$ is selected as the one minimizing $\mathcal{J}(1, h)$ (Step 2). Mode $s(t)$ at time $t \geq 2$ is then selected as the one minimizing the "cost-to-go" $\mathcal{J}(t, h)$ over $h \in \mathcal{S}$ (Step 3.3), with $\mathcal{J}(t, h)$ obtained as in Step 3.2. Note that, $\mathcal{J}(t, h)$ is computed recursively based on the cost-to-go at the previous time step $\mathcal{J}(t-1, d)$, $d \in \mathcal{S}$, and the current aggregate measurement $y(t)$. Consequently, Algorithm 2 does not require to store past data and it is thus suited for real-time disaggregation.

Algorithm 2 Mode sequence inference

Input: Aggregate output readings flow $y(1), y(2), \dots$; model parameters Θ_i, $i = 1, \dots, N$; parameters $\lambda(d)$, $d \in \mathcal{S}$.

1. $\mathcal{J}(1, h) \leftarrow \eth(y(1), h)$; $h \in \mathcal{S}$;
2. $s(1) \leftarrow \arg\min_{h \in \mathcal{S}} \mathcal{J}(1, h)$;
3. **iterate for** $t = 2, \dots$
   3.1. $\mathcal{J}'(t) \leftarrow \min_{d \in \mathcal{S}} \left(\mathcal{J}(t-1, d) + \lambda(d)\mathbf{1}(h \neq d)\right)$;
   3.2. $\mathcal{J}(t, h) \leftarrow \eth(y(t), h) + \mathcal{J}'(t)$, $h \in \mathcal{S}$;
   3.3. $s(t) \leftarrow \arg\min_{h \in \mathcal{S}} \mathcal{J}(t, h)$;

Output: Estimated joint mode sequence $s(1), s(2), \dots$

5. NUMERICAL RESULTS

The performance of the approach described in this paper is assessed on real consumption data taken from the AMPDs dataset (Makonin et al., 2016). This dataset consists of the electric power consumption readings of 19 devices taken at 1-minute time resolution in a single house located in Canada. We only model 5 out of 19 appliances, namely, cloth dryer (CDE); dishwasher (DWE); fridge (FGE); heat pump (HPE); basement plugs & lights (BME).

5.1 Single appliance modelling

Part of the AMPDs dataset is used to model each of the five appliances. In particular, the training sets $\{\mathcal{M}_i\}_{i=1}^5$ consist of one week of power consumption records ($\bar{T} = 10080$ samples) for each device. To have informative data sets, we consider different weeks for different appliances. In particular, readings from day 74 to day 80 form the training set for the clothes dryer, from day 170 to day 176 for the dishwasher, from day 23 to day 29 for fridge, heat pump and basement plugs & lights.

Models for each appliance are estimated running Algorithm 1. The hyper-parameters λ_i, with $i = 1, \dots, 5$, are all set to 0.9. As Algorithm 1 may trap in a local minimum, it is run for 10 different randomly generated initial sequences S_i^0, and the estimate providing the minimum value for the cost J_i is selected. The number of sub-models K_i is chosen through cross-validation and it results in: $K_1 = 3$ (CDE), $K_2 = 3$ (DWE), $K_3 = 2$ (FGE), $K_4 = 2$ (HPE) and $K_5 = 2$ (BME).

The estimated parameters Θ_i for each appliance are reported in Table 1. Note that, due to the chosen model class (2), the parameters represent the estimated power consumption of each appliance at the different operating regimes.

The performance of the trained models are assessed on the following one-day ($T_{\text{v}} = 1440$ samples) validation sets: day 170 for CDE ($\mathcal{V}_1$), day 45 for DWE ($\mathcal{V}_2$), day 234 for FGE ($\mathcal{V}_3$), day 80 for HPE ($\mathcal{V}_4$) and day 300 for BME ($\mathcal{V}_5$). The

Table 1. Appliances modelling. Estimated parameters Θ_i for each appliance.

Appliance	Θ_i
CDE	$\begin{bmatrix}0.4 & 251.8 & 4644.3\end{bmatrix}$
DWE	$\begin{bmatrix}0.1 & 781.2 & 146.6\end{bmatrix}$
FGE	$\begin{bmatrix}1.1 & 132.8\end{bmatrix}$
HPE	$\begin{bmatrix}33.7 & 1822.1\end{bmatrix}$
BME	$\begin{bmatrix}5.7 & 330.5\end{bmatrix}$

Table 2. Appliances modelling. Achieved BFR for the five estimated models on the validation sets $\{\mathcal{V}_i\}_{i=1}^5$.

Model / Set	CDE	DWE	FGE	HPE	BME
$\mathcal{V}_1$ (CDE)	95.5%	15.8%	2.2%	37.1%	6.3 %
$\mathcal{V}_2$ (DWE)	30.5%	98.0%	15.2%	0%	10.6 %
$\mathcal{V}_3$ (FGE)	0%	81.8%	91.4%	0%	0%
$\mathcal{V}_4$ (HPE)	10.0%	40.1%	3.3%	91.3%	14.7%
$\mathcal{V}_5$ (BME)	72.6 %	38.5 %	34.1 %	0%	97.1 %

quality of the estimated models is quantified in terms of *Best Fit Rate* (BFR)

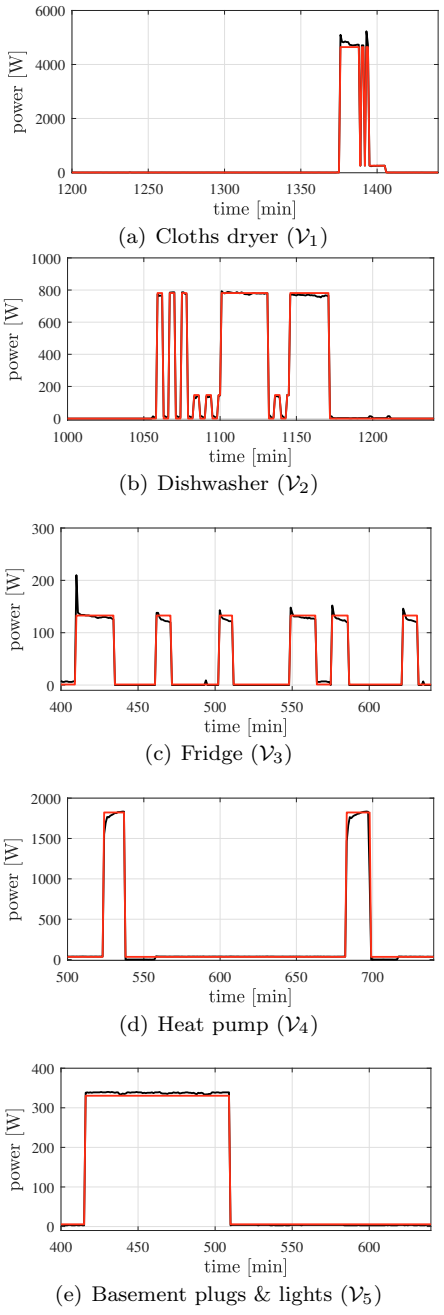
$$\text{BFR}_i = 100 \max\left\{1 - \frac{\sqrt{\sum_{t=1}^{T_v}\left(y_i(t) - \hat{y}_i(t)\right)^2}}{\sqrt{\sum_{t=1}^{T_v}\left(y_i(t) - \bar{y}_i\right)^2}}, 0\right\}\%, \tag{11}$$

with $\bar{y}_i$ denoting the sample mean of the output and $\hat{y}_i(t)$ being the output predicted by the model. The BFRs shown in Table 2 are computed testing all the models over all the available validation sets $\mathcal{V}_i$, $i = 1, \dots, 5$. It can be noticed that the BFRs in the diagonal entries of the table are always higher than 90% and they are the highest one of the corresponding row. This basically means that the estimated models provide useful information in detecting which appliance has generated a given power consumption pattern. Note that the BFR achieved by the model of the dishwasher (DWE) is relatively high (81.8%) on the power consumption pattern of the fridge (FGE). Furthermore, on the validation set $\mathcal{V}_5$, a BFR of 72.6% is achieved by the model of the clothes dryer (CDE), even if $\mathcal{V}_5$ comprises the power readings of the basement plugs & lights (BME). These results are mainly due to some similarities in the consumption patterns of those appliances, that can be observed comparing the estimated parameters Θ_i for the DWE and the FGE model and the ones for the CDE and BME model (see Table 1).

To further assess the quality of the estimated models, in Fig. 1 we compare the actual and the predicted outputs obtained in validation. For a better visualization, only the results on 240 samples are reported. It can be seen that the behaviour of each appliance is accurately described by the corresponding estimated jump model.

5.2 Energy disaggregation

Capabilities of the disaggregation algorithm in reconstructing the appliances' power consumption from the aggregate readings are now assessed. The considered disaggregation dataset $\mathcal{D}_T$ is disjoint from the sets $\mathcal{M}_i$,



(a) Cloths dryer ($\mathcal{V}_1$)

(b) Dishwasher ($\mathcal{V}_2$)

(c) Fridge ($\mathcal{V}_3$)

(d) Heat pump ($\mathcal{V}_4$)

(e) Basement plugs & lights ($\mathcal{V}_5$)

Fig. 1. Single appliance modelling. True (black) vs estimated (red) consumption patterns.

$i = 1, \dots, 5$, and it contains measurements over 12 consecutive days ($T = 17280$). The aggregate measurements $y(t)$ on the dataset $\mathcal{D}_T$ are synthetically constructed by summing up the power consumption of the five considered appliances. To assess robustness against modelling errors, we also corrupt the obtained aggregated power $y(t)$ with a fictitious zero-mean Gaussian noise with standard deviation 4 W and then we add the unmodelled consumption patterns of bedroom, garage and dining room on top of $y(t)$. We remark that the end-use profiles available from the AMPDs dataset are only used as ground-truth when evaluating the performance of the disaggregation method.

To quantitatively assess the performance of the disaggregation approach we use the following metrics:

1. The F-score Fs (Batra et al., 2014):
$$Fs_i = 2\frac{PC_i \times RC_i}{PC_i + RC_i}. \quad (12)$$
The indexes RC_i and PC_i in (12) are defined as
$$RC_i = \frac{TP_i}{TP_i + FN_i}, \quad PC_i = \frac{TP_i}{TP_i + FP_i},$$
where TP_i, FP_i and FN_i are the true positives (correctly classified on events), the false positives (off events labeled as on) and false negatives (on events classified as off), respectively.

   The F-score provides an indication of the capabilities of the disaggregation method in detecting the appliance on/off states. As the estimated models are not limited to the description of on/off events, we classify the cloths dryer, dishwasher, fridge and basement as off when their power consumption is less than 10 W. This threshold is set to 50 W for the heat pump, as its estimated low-level power consumption is 33.7 W (see Table 1).
2. The *Estimated Energy Fraction Index* (EEFI):
$$\text{EEFI}_i = \frac{\sum_{t=1}^{T} \hat{y}_i(t)}{\sum_{i=1}^{N}\sum_{t=1}^{T} \hat{y}_i(t)}. \quad (13)$$
The EEFI_i index represents the fraction of energy assigned to the i-th appliance based on the reconstructed consumption pattern. It is compared with the *Actual Energy Fraction Index* (AEFI):
$$\text{AEFI}_i = \frac{\sum_{t=1}^{T} y_i(t)}{\sum_{i=1}^{N}\sum_{t=1}^{T} y_i(t)}. \quad (14)$$
which indicates the actual fraction of energy consumed by the i-th appliance.
3. The *Relative Square Error* (RSE):
$$\text{RSE}_i = \frac{\sum_{t=1}^{T}(y_i(t) - \hat{y}_i(t))^2}{\sum_{t=1}^{T} y_i^2(t)}. \quad (15)$$
4. The R^2 coefficient
$$R_i^2 = 1 - \frac{\sum_{t=1}^{T}(y_i(t) - \hat{y}_i(t))^2}{\sum_{t=1}^{T}(y_i(t) - \bar{y}_i)^2}, \quad (16)$$
where $\bar{y}_i$ is the sample mean of $\{y_i(t)\}_{i=1}^{T}$.

   Both R_i^2 and RSE_i measure the quality of the estimated consumption pattern for the i-th device over time.

The values of the performance metrics achieved by Algorithm 2 are reported in Tables 3 and 4, while the reconstructed single-appliance consumption trajectories over time are plotted in Fig. 2. A comparison between AEFI_i and EEFI_i, with $i = 1, \dots, 5$ indicates that the disaggregation method enables to accurately estimate the fraction of energy consumed by each appliance. This result is related to the accuracy of the disaggregated trajectories (see the RSE_i and the R_i^2 indexes in Table 4). Although the resulting RSE_i and R_i^2 seem to indicate that the mismatch between actual and estimate trajectory over time for the fridge is not negligible, the estimated profile plotted in Fig. 2 accurately reproduces the behaviour of the fridge most of the times. The obtained RSE_i and R_i^2 are thus strongly influenced by the errors in the reconstructed consumption of the fridge. These mismatches are due to the unmodelled transient of this appliance and the similarities between the high-power and the mid-power levels of the fridge and the dishwasher, respectively.

Table 3. Energy disaggregation. Actual Energy Fraction Index AEFI_i vs Estimated Energy Fraction Index EEFI_i.

	AEFI_i	EEFI_i
Clothes dryer	12%	14.3%
Dishwasher	2.2%	4.1%
Fridge	8%	6.8%
Heat Pump	66.3%	66.7%
Basement	11.5%	8.1%

Table 4. Energy disaggregation. Achieved $F-$scores (Fs_i), Relative Square Errors (RSE_i) and R_i^2 coefficients.

	Fs_i	RSE_i	R_i^2
Clothes dryer	97.8%	15.7%	84.1%
Dishwasher	95.6%	37.0%	62.4%
Fridge	94.4%	42.5%	39.6%
Heat Pump	99.9%	1.3%	98.3%
Basement	96.4%	34.3%	56.6%

All computations are carried out on an i7 2.80-GHz Intel core processor with 16 GB of RAM running MATLAB R2016b. The average CPU time required to disaggregate the total power consumed at a given time instant is 0.25 ms.

6. CONCLUDING REMARKS AND FUTURE WORK

In this paper we have applied the approach for jump model fitting and filtering recently proposed by (Bemporad et al., 2017) to: (i) model consumption patterns of residential electric appliances and (ii) decompose, in real-time, the total power consumption measured at a household level into the individual contributions of the single appliances.

Current research addresses the identification of dynamic sub-models to better characterize the transient behaviour of the single appliances and integration of single jump model identification into the disaggregation procedure, in order to simultaneously reconstruct the disaggregated signals and update the models of the single devices.

REFERENCES

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

Bemporad, A., Breschi, V., Piga, D., and Boyd, S. (2017). Fitting jump models. `https://arxiv.org/abs/1711.09220`.

Cominola, A., Giuliani, M., Piga, D., Castelletti, A., and Rizzoli, A. (2017). A hybrid signature-based iterative disaggregation algorithm for non-intrusive load monitoring. *Applied Energy*, 185(P1), 331–344.

Esa, N., Abdullah, P., and Hassan, M. (2016). A review disaggregation method in non-intrusive appliance load monitoring. *Renewable and Sustainable Energy Reviews*, 66, 163–173.

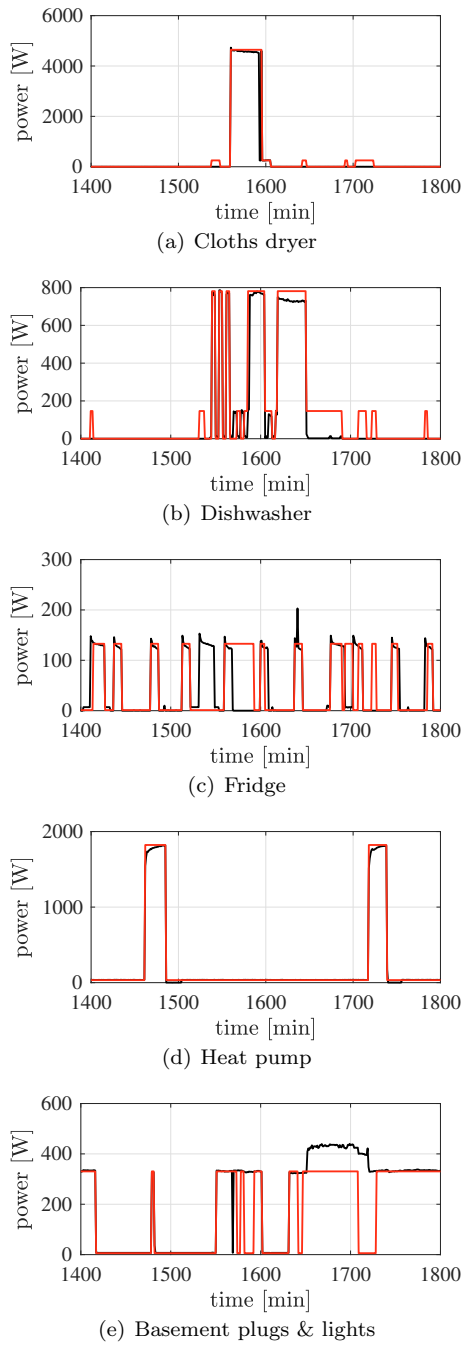


Fig. 2. Energy disaggregation results. True (black) vs reconstructed (red) consumption patterns.

Faustine, A., Mvungi, N.H., Kaijage, S., and Michael, K. (2017). A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem. *CoRR*. URL `http://arxiv.org/abs/1703.00785`.

Hart, G.W. (1992). Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12), 1870–1891.

Kolter, J.Z. and Jaakkola, T. (2012). Approximate inference in additive factorial hmms with application to energy disaggregation. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22, 1472–1482.

Makonin, S., Bradley, E., Bajic, I., and Popowich, F. (2016). Electricity, water, and natural gas consumption of a residential house in canada from 2012 to 2014. *Scientific Data*, 3(160037), 1–12.

Piga, D., Cominola, A., Giuliani, M., Castelletti, A., and Rizzoli, A.E. (2016). Sparse optimization for automated energy end use disaggregation. *IEEE Transactions on Control Systems Technology*, 24(3), 1044–1051.

Suzuki, K., Inagaki, S., Suzuki, T., Nakamura, H., and Ito, K. (2008). Nonintrusive appliance load monitoring based on integer programming. In *2008 SICE Annual Conference*, 2742–2747.